

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-263440

(43)Date of publication of application : 11.10.1996

(51)Int.Cl. G06F 15/00
G06F 12/00
G06F 17/60
G09C 1/00
H04L 9/32

(21)Application number : 07-299840

(71)Applicant : XEROX CORP

(22)Date of filing : 17.11.1995

(72)Inventor : MARK J STEFFEK

(30)Priority

Priority number : 94 344041

Priority date : 23.11.1994

Priority country : US

(54) DIGITAL WORK DISTRIBUTION SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a control system of the use and distribution of a digital work.

SOLUTION: A right on using right grammar is a label corresponding to a prescribed action and conditions for the exercise of the right. A copyright indicates that the digital work can be copied and the condition is that a requester delivers a specified security standard. A loanright is defined so as to limit the loan period of the work. A using right is provided with a copy count condition for limiting the number of times that the right is simultaneously exercised, a security class condition for ensuring that the requester is provided with the security of an appropriate level, a time specification for indicating the restriction condition of a time base for the exercise of the right and the right code of charge use or the like for indicating a using charge for the exercise of the right along with the condition for the exercise of the right.

THIS PAGE BLANK (USPTO)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-263440

(43) 公開日 平成8年(1996)10月11日

(51) Int.Cl. ⁸	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 15/00	3 3 0	9364-5L	G 0 6 F 15/00	3 3 0 Z
12/00	5 3 7	7623-5B	12/00	5 3 7 M
17/60		7259-5J	G 0 9 C 1/00	6 6 0 D
G 0 9 C 1/00	6 6 0	7259-5J		6 6 0 E
			G 0 6 F 15/21	Z

審査請求 未請求 請求項の数 2 O L (全 40 頁) 最終頁に続く

(21) 出願番号 特願平7-299840

(22) 出願日 平成7年(1995)11月17日

(31) 優先権主張番号 3 4 4 0 4 1

(32) 優先日 1994年11月23日

(33) 優先権主張国 米国 (U S)

(71) 出願人 590000798

ゼロックス コーポレイション

XEROX CORPORATION

アメリカ合衆国 ニューヨーク州 14644

ロチェスター ゼロックス スクエア

(番地なし)

(72) 発明者 マーク ジェイ. ステフィック

アメリカ合衆国 94062 カリフォルニア

州 ウッドサイド ビッグ ツリー ウェ

イ 55

(74) 代理人 弁理士 中島 淳 (外1名)

(54) 【発明の名称】 デジタルワーク配給システム

(57) 【要約】

【課題】 デジタルワークの使用及び配給の制御システムを提供する。

【解決手段】 使用権文法上の権利は、権利の行使のための所定の行動と条件に対応するラベルであり、COPY RIGHT (著作権) はデジタルワークがコピーされてもよいことを示し、その条件はリクエストが特定のセキュリティ基準を渡すことであり、LOAN RIGHT (貸出権) はワークの貸出し期間を制限するために定義される。使用権は、権利の行使のための条件と共に、権利が同時に行われる回数を制限するコピーカウント条件、リクエストが適切なレベルのセキュリティを有していることを確認とするセキュリティクラス条件、アクセステストを指定するアクセス条件、権利行使のためのタイムベースの制約条件を示すタイム仕様、権利の行使に対する使用料を示す料金使用等の権利コードを含む。

```

1501 ~ Digital Work Rights := (Rights*)
1502 ~ Right := (Right-Code (Copy-Count) (Control-Spec) (Time-Spec)
(Access-Spec) (Fee-Spec))
1503 ~ Right-Code := (Render-Code (Transport-Code) (File-Management-
Code) (Derivative-Work-Code) (Configuration-Code))
1504 ~ Render-Code := (Play (Player-Player-ID) (Print (Printer-Printer-ID))
1505 ~ Transport-Code := (Copy (Transfer (Lane (Remaining-Right)
Next-Set-of-Right) (New-Copy-Right) (Next-Set-of-Right))
1506 ~ File-Management-Code := (Backup (Back-Up-Copy-Right)
Next-Set-of-Right) (Restore (Delete (Folder
Directory (Name: Hide-Local (Hide-Remote)
(Purge: Hide-Local (Hide-Remote)
1507 ~ Derivative-Work-Code := (Extract (Embed (Edit (Process:
Process-ID) (Next-Copy-Right)
Next-Set-of-Right)
1508 ~ Configuration-Code := (Install (Uninstall)
1509 ~ Next-Set-of-Right := (Add (Set-Of-Right) (Delete:
Set-Of-Right) (Replace Set-Of-Right) (Keep Set-Of-Right))
1510 ~ Copy-Count := (Copies: positive integer) (0 (Unlimited))
1511 ~ Control-Spec := (Control (Restrictable) (Unrestrictable)
(Unchargeable) (Chargeable))
1512 ~ Time-Spec := (Time-Interval (Scheduling-Interval) (Meter-Time)
Until: Expiration-Date)
1513 ~ Fixed-Interval := From: Start-Time
1514 ~ Scheduling-Interval := (Interval) (Use-Duration)
1515 ~ Meter-Time := Time-Restaining: Rescheduling-Use
1516 ~ Access-Spec := ((SQ: Security-Class) (Authorization: Authorization-ID)
(Other-Authorization: Authorization-ID) (Ticket: Ticket-ID))
1517 ~ Fee-Spec := (Scheduled-Discount) (Regular-Fee-Spec) (Scheduled-Fee-Spec)
(Markup-Spec)
1518 ~ Scheduled-Discount := (Schedule & Discount) (Scheduled-Discount:
(Time-Spec Percentage)*)
1519 ~ Regular-Fee-Spec := (Fee (Increment) (Flat-Fee-Spec) (Metered-Rate-
Spec) (Best-Price-Spec) (Call-For-Price-Spec)
(Offa: Money-Unit Per: Time-Spec) (Max:
Money-Unit Per: Time-Spec) To: Account-ID)
1520 ~ Flat-Fee-Spec := Per: Use: Money-Unit
1521 ~ Metered-Rate-Spec := Metered: Money-Unit Per: Time-Spec
1522 ~ Best-Price-Spec := Best-Price: Money-Unit Max: Money-Unit
1523 ~ Call-For-Price-Spec := Call-For: Price
1524 ~ Scheduled-Fee-Spec := (Schedule (Time-Spec Regular-Fee-Spec)*)
1525 ~ Markup-Spec := Markup: percentage To: Account-ID

```

【特許請求の範囲】

【請求項1】 アタッチされた一つ以上の使用権を有するデジタルワークを配給するための配給システムであって、

関連するデジタルワークの所有者が前記関連するデジタルワークを伝送する方法を示す使用権のインスタンスを作成する文法を備え、

前記文法から使用権を作成する手段を備え、

作成された使用権をデジタルワークへアタッチする手段を備え、

デジタルワークへアクセスするためのリクエストリポジトリを備え、前記要求リポジトリが使用トランザクションを発生する手段を有し、前記使用トランザクションの各々が使用権を指定し、

アタッチされた作成された使用権を有するデジタルワークを記憶するサーバリポジトリを備え、前記サーバリポジトリが、前記リクエストリポジトリからの使用トランザクションを処理してデジタルワークへのアクセスが許諾されるか否かを決定する手段を有していることよりなるデジタルワーク配給システム。

【請求項2】 デジタルワークの配給及び使用を制御するためのコンピュータベースシステムであって、

デジタルワークがいかにして使用され又は配給されるかを定義する使用権のインスタンスを作成する使用権文法を備え、前記使用権文法が使用権の伝送を定義するための第1の複数の文法エレメントと使用権のレンダリングを定義するための第2の複数の文法エレメントとを有し、

使用権をデジタルワークへアタッチする手段を備え、デジタルワークを記憶すると共に交換するための複数のリポジトリを備え、前記複数のリポジトリの各々が、デジタルワークとそれらのアタッチされた使用権を記憶する手段と、

トランザクション処理手段であって、要求されたデジタルワークへのアクセスを要求するためのリクエスト動作モードを有し、前記要求が使用権を指定し、前記要求において指定された使用権と前記要求されたデジタルワークへアタッチされた使用権とに基づいて、前記要求されたデジタルワークへアクセスするための要求を処理するためのサーバ動作モードを有する、前記トランザクション処理手段と、

通信媒体を介して前記複数のリポジトリの他のリポジトリと結合する結合手段と、を有することよりなるデジタルワークの配給及び使用の制御用コンピュータベースシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、デジタルコード化ワーク（作品、著作物）の配給及び使用権実施の分野に関する。

【0002】

【従来の技術及び発明が解決しようとする課題】 電子出版と見なされる出版（パブリッシング）及び情報産業が直面している基本的な問題は、電子出版された資料（データ）の無許可、料金請求されない配給又は使用を如何にして阻止するかにある。電子出版される資料は、一般的にデジタル形式で配給され、これらの資料を再作成する（recreate）能力を有するコンピュータベースシステムにおいて再作成される。オーディオ及びビデオレコーディング、ソフトウェア、書籍、並びにマルチメディアワークは、全て電子出版されている。これらの業界における企業は、小売店におけるオーディオCDの販売のような資料のデリバリについて責任を負うことに対するローヤリティを受け取る。ワーク（作品、著作物、著書を全般的に称する）の料金請求されない配給は結果的に支払われないローヤリティを生じることになる（例えば、オーディオレコーディングCDを他のデジタル媒体へコピーすること）。

【0003】 電子出版されたワークが、「完全に」再生され、配給されることができるとの簡単さが主な問題である。ネットワーク全体におよぶデジタルワークの転送は今や常識である。このように広範囲に使用される一つのネットワークは、Internet（インターネット）である。インターネットは、多くの大学、企業、及び政府機関におけるコンピュータのユーザが通信し、アイデアや情報を交換する広範囲なネットワークファシリティである。CompuServe及びProdiqyのようなインターネット及び商業ネットワーク上に設けられたコンピュータ掲示板は、デジタル情報のポスティング及び検索を許容する。Dialog及びLEXIS/NEXISのような情報サービスは、幅広いトピックに対して現在情報のデータベースを提供する。さらに状況を悪化させる他のファクタは、NII（米国における全国情報基盤整備政策）の開発と拡大である。NIIが成長するにつれて、ネットワーク全体にわたるデジタルワークの転送が何回でも繰り返されて増大することが予想される。従って、広範囲な無許可コピーの危険性を伴わずに、デジタルワークの配給のためにNIIを使用することが所望される。

【0004】 料金請求されない配給を抑制する最も簡単な方法は、無許可コピー及び転送を防止することにある。デジタル形式で配給された現存の資料に関しては、種々の保護手段が使用されている。ソフトウェアの場合、作成され得るコピー数を制限したり又はコピー（複製）が検出された時に出力した物を改悪するようなコピー保護方式（スキーム）が用いられてきた。他の方式は、所定期間を経過すると、ソフトウェアを動作禁止にする。ワークステーションベースのソフトウェアに使用される技術は、ソフトウェアをランするには特別なハードウェアデバイスがワークステーション上に存在すべきであることを要求する。これに関しては、例えば、M

ethod and Apparatus for Protecting Computer Software Utilizing Coded Filter Network in conjunction with an Active Coded Hardware Device”と題された米国特許第4,932,054号を参照されたい。このような装置は、一般に、“dongles”と呼ばれるソフトウェアを備えている。

【0005】さらに他の方式は、ソフトウェアを動作可能にするための「キー」を必要とするソフトウェアを配給することである。これは、ソフトウェアの“demos”（デモンストレーションの意）がプロダクト全体と共に媒体上に提供されるような配給方式において使用される。demos は自由に使用されることができ、実際のプロダクトを使用する為には、キーが購入されなければならない。まず最初にキーを購入すれば、これらの方式はソフトウェアのコピーを妨害しない。

【0006】

【課題を解決するための手段】デジタルワークの使用及び配給を制御するシステムが開示されている。デジタルワークは、デジタル形式で変換されるか又は作成されると共にソフトウェアプログラムのような適切なレンダリング手段を用いて再作成され得るあらゆる書き込み、聴覚、グラフィカル、又はビデオベースのワークである。一つ以上のデジタルワークを含むフォルダは、デジタルワークとして処理されることができ、本発明はデジタルワークのオーナーが彼らのワークに使用権をアタッチするのを可能にする。これらの使用権はデジタルワークがいかんして使用され、配給されるかを定義する。これらの使用権はデジタルワークの一部となり常に遵守される。

【0007】使用権のインスタンスはフレキシブルで拡張可能な使用権文法を用いて定義される。概念的には、使用権文法における権利は権利を行使することに対する所定の行動と条件に対応したラベルである。例えば、COPY right（著作権）はデジタルワークのコピーが行われてもよいことを表記する。この権利の行使に対する条件は、リクエストがいくらかのセキュリティ（機密保護）基準を渡さなければならない事にある。条件は権利其れ自体を制限するためにもアタッチされる。例えば、LOAN right（貸出し権）はワーク（作品）が貸し出される期間を制限するために定義されてもよい。

【0008】本発明においては、使用権は、権利を行使するための種々の条件と一緒に権利コードを備えている。このような条件は、権利が同時に行使される回数を制限するコピーカウント（複製回数）の条件（例えば、所定回数ローンでのコピー数制限）、リクエストが適切なレベルのセキュリティを有していることを確認とするセキュリティクラス条件、渡されるべきアクセステストを指定するためのアクセス条件、権利を行使するためのタイムベースの制約条件を示すタイム仕様、及び権利の行使に対する使用料を示す料金仕様を含む。デジタル

ワークはデジタルワークへアタッチされた権利の異なるバージョンを有することができる。権利のバージョンは他のバージョンと同じ権利コードを有するが、条件（及び一般的には料金）は異なる。

【0009】デジタルワークとそれらのアタッチされた使用権はリポジトリ内に記憶される。デジタルワークはリポジトリ同士の間で転送される。リポジトリは使用トランザクションステップの所定のセットに従ってデジタルワーク同士を交換するように対話する。使用権の行動は使用トランザクションステップの所定のセット内に組み込まれる。当該使用トランザクションステップは権利が行使される前に満たされるべき全ての条件を更にチェックする。これらの使用権トランザクションステップは権利の行使の要求と権利の実行のためにリポジトリによって使用されるプロトコルを定義する。

【0010】本発明の一つの態様は、アタッチされた一つ以上の使用権を有するデジタルワークを配給するための配給システムであって、関連するデジタルワークの所有者が前記関連するデジタルワークを伝送する方法を示す使用権のインスタンスを作成する文法を備え、前記文法から使用権を作成する手段を備え、作成された使用権をデジタルワークへアタッチする手段を備え、デジタルワークへアクセスするためのリクエストリポジトリを備え、前記要求リポジトリが使用トランザクションを発生する手段を有し、前記使用トランザクションの各々が使用権を指定し、アタッチされた作成された使用権を有するデジタルワークを記憶するサーバリポジトリを備え、前記サーバリポジトリが、前記リクエストリポジトリからの使用トランザクションを処理してデジタルワークへのアクセスが許諾されるか否かを決定する手段を有していることよりなるデジタルワーク配給システムである。

【0011】本発明の他の態様は、デジタルワークの配給及び使用を制御するためのコンピュータベースシステムであって、デジタルワークがいかんして使用され又は配給されるかを定義する使用権のインスタンスを作成する使用権文法を備え、前記使用権文法が使用権の伝送を定義するための第1の複数の文法エレメントと使用権のレンダリングを定義するための第2の複数の文法エレメントとを有し、使用権をデジタルワークへアタッチする手段を備え、デジタルワークを記憶すると共に交換するための複数のリポジトリを備え、前記複数のリポジトリの各々が、デジタルワークとそれらのアタッチされた使用権を記憶する手段と、トランザクション処理手段であって、要求されたデジタルワークへのアクセスを要求するためのリクエスト動作モードを有し、前記要求が使用権を指定し、前記要求において指定された使用権と前記要求されたデジタルワークへアタッチされた使用権とに基づいて、前記要求されたデジタルワークへアクセスするための要求を処理するためのサーバ

動作モードを有する、前記トランザクション処理手段と、通信媒体を介して前記複数のリポジトリの他のリポジトリと結合する結合手段と、を有することよりなるデジタルワークの配給及び使用の制御用コンピュータベースシステムである。

【0012】

【発明の実施の形態】 デジタルワークの使用及び配給を制御するためのシステムが開示されている。本発明はデジタルワークを含む商業的トランザクション（取引）をサポートすることに関する。

【0013】 本明細書中において、用語「デジタルワーク」、「ワーク」及び「内容」は、デジタル表現に縮小されたあらゆるワーク（作品、著作物、著書等）を称する。これは、あらゆるオーディオ、ビデオ、テキスト又はマルチメディアワーク、及びワークを再作成するために必要とされ得るあらゆる付随するインタープリタ（例えば、ソフトウェア）を含む。用語「複合ワーク」は、他のデジタルワークの集合からなるデジタルワークを称する。用語「使用権」又は「権利」は、デジタルワークの配布先として許諾された権利を称する用語である。一般に、これらの権利は、デジタルワークがいかんして使用され得るか、及び当該デジタルワークが更に配給され得るか否かを定義する。各使用権は、権利が行使される前に満たされなければならない一つ以上の指定条件を有している。

【0014】 図1は、種々の詳細は省略しているが、本発明の基本的な動作を示す高レベルフローチャートである。図1に関しては、ステップ101において、クリエイターがデジタルワークを作成する。ステップ102において、クリエイターは適切な使用権及び料金を決定し、デジタルワークにそれらをアタッチし、次いでそれらをリポジトリ1内に記憶する。適切な使用権及び料金の決定は様々な経済的ファクタによる。アクセス要求が受信されるまで、デジタルワークはリポジトリ1内に安全に残されている。アクセス要求は、他のリポジトリによるセッション開始から始まる。この時、ステップ103において、リポジトリ2は、リポジトリ1とセッションを開始する。以下により詳細に記述されているように、このセッション開始は、それぞれのリポジトリが信頼できることを確実とするのを助けるステップを含む。セッションが設定されることができると仮定すると、ステップ104において、リポジトリ2は主張された目的のためにDigital Work（デジタルワーク）へのアクセスを要求することができる。この目的は、例えば、デジタルワークをプリントすること又はデジタルワークのコピーを入手することであってもよい。この目的は、特定の使用権に対応している。いかなる場合でも、ステップ105において、リポジトリ1はデジタルワークへのアクセスが許諾されたか否かを決定するためにデジタルワークに対応する使用権をチェックする。使用権

のチェックは、本質的に、アクセス要求に対応している権利がデジタルワークにアタッチされているか否か、及びこの権利に対応している全ての条件が満たされているか否かについての決定を含んでいる。アクセスが拒絶された場合、ステップ106において、リポジトリ1はエラーメッセージによってセッションを終了する。アクセスが許諾された場合、ステップ107において、リポジトリ1はデジタルワークをリポジトリ2へ転送する。デジタルワークがリポジトリ2へ転送されると、

10 ステップ108において、リポジトリ1及び2はそれぞれアクセスに対して請求情報を発生し、当該請求情報はクレジットサーバへ転送される。このような2重の請求報告は、請求処理を不正に欺こうとする企てから守るために行われる。

【0015】 図2は、本発明におけるリポジトリタイプ同士の基本的な対話を示す。図2から明らかになるように、種々のリポジトリタイプは異なる機能を提供する。安全であると共に信頼できる通信を可能とする機能のコアセットをリポジトリが共有することが基本であ

20 る。図2に関しては、リポジトリ201はリポジトリの一般的なインスタンスを示す。リポジトリ201は、サーバモードとリクエストモードの二つの動作モードを有している。サーバモードの時、リポジトリは、デジタルワークへのアクセス要求を受け取ったり、処理したりする。リクエストモードの時、リポジトリは、デジタルワークへアクセスするための要求を開始する。リポジトリ201は、その主な目的がデジタルワークのための交換媒体である点で一般的である。動作の過程において、当該リポジトリ201は、複数の他のリポジトリ、

30 即ち、オーソライゼーションリポジトリ202、レンダリングリポジトリ203及びマスタリポジトリ204と通信してもよい。リポジトリ同士の間の通信は、リポジトリ・トランザクション・プロトコル205を用いて行われる。

【0016】 オーソライゼーションリポジトリ202との通信は、アクセスされているデジタルワークがオーソライゼーションを要求する条件を有している時に行われる。概念的に、オーソライゼーションとは、デジタル的証明書であり、デジタルワークへのアクセスを得る為にこの証明書の所有が要求される。オーソライゼーションとは、リポジトリ同士の間で移動できると共に料金や使用権の条件に当てられるデジタルワークそれ自体である。オーソライゼーションは、デジタルワークへのアクセスにおいて関係する両リポジトリによって要求されてもよい。

【0017】 レンダリングリポジトリ203との通信は、デジタルワークのレンダリングに関して行われる。以下に詳細に記述されるように、レンダリングリポジトリはレンダリングシステムを備えるレンダリングデバイスと連結されている（例えば、プリンタデバイ

50

ス)。

【0018】マスタリポジトリ205との通信は、識別証明書を入手することに関して行われる。識別証明書は、リポジトリが「信頼できる」として識別されるための手段である。識別証明書の使用は、登録トランザクションに関して以下に記述されている。

【0019】図3は、クレジットサーバ301に連結されたリポジトリ201を示す。クレジットサーバ301は、リポジトリ201に対する請求情報を累積するデバイスである。このクレジットサーバ301は、請求トランザクションを記録するために請求トランザクション302を介してリポジトリ201と通信する。請求トランザクションは、クレジットサーバによって請求クリアリングハウス（情報機関）303へ定期的に報告される。クレジットサーバ301は、クリアリングハウストランザクション304を介して請求クリアリングハウス303と通信する。クリアリングハウス・トランザクション304は請求クリアリングハウス303への情報の安全な転送、即ち暗号化された転送を可能とする。

【0020】レンダリングシステムは、リポジトリと、所望される形式でデジタルワークをレンダリングすることができるレンダリングデバイスとを備えたシステムとして一般に定義される。レンダリングシステムの実施例は、コンピュータシステム、デジタルオーディオシステム、又はプリンタであってもよい。レンダリングシステムはリポジトリと同じセキュリティ（機密保護）特性を有している。レンダリング・リポジトリのレンダリングデバイスとの連結はレンダリングデバイスのタイプに適した方法で行われてもよい。

【0021】図4(a)は、レンダリングシステムの実施例としてのプリンタを示している。図4に関しては、プリンタシステム401は、プリンタリポジトリ402及びプリンタデバイス403を含んでいる。なお、当該プリンタシステム401を画定するダッシュラインは、セキュリティシステム境界を画定している。境界との通信は安全であると仮定されている。セキュリティレベルによって、境界は、物理的な完全性を提供するように意図されたバリアも示す。プリンタリポジトリ402は、図2のレンダリングリポジトリ205のインスタンス生成である。当該プリンタリポジトリ402は、幾つかのインスタンスにおいて、プリンタエンジン403によってプリントアウトされるまで残るデジタルワークの一回切りのコピーも含んでいる。他のインスタンスにおいては、当該プリンタリポジトリ402は、使用に基づいて残ったり請求されることができるフォントのようなデジタルワークを含む。このデザインは、プリンタとプリント装置との間の全通信ラインが、物理的に安全な境界の中に入らない場合に、暗号化されることを確実とする。このデザイン形態によって、デジタルワークが不正に得られるような潜在的な「フォルト」ポイントが取り除

かれる。プリンタデバイス403は、プリントされた出力を作成するために使用されるプリンタ構成要素を示す。

【0022】図4の(a)にはリポジトリ404も示されている。当該リポジトリ404はプリンタリポジトリ402に連結されている。当該リポジトリ404はデジタルワークを含む外部リポジトリを示す。

【0023】図4の(b)は、レンダリングシステムとしてのコンピュータシステムの実施例である。コンピュータシステムは、デジタルワーク（例えば、ソフトウェアプログラム）を実行(execute)し、デジタルワーク（例えば、デジタル化された写真）をディスプレイするので、「多機能(multi-function)」デバイスを備えていてもよい。論理的には、物理的リポジトリを一つだけしか必要としないにもかかわらず、各レンダリングデバイスはそれ自体のリポジトリを有しているように見える。図4の(b)に関しては、コンピュータシステム410は、ディスプレイ/実行リポジトリ411を含んでいる。当該ディスプレイ/実行リポジトリ411は、ディスプレイデバイス412及び実行デバイス413に連結されている。当該コンピュータシステム410を囲むダッシュラインボックスは、通信が安全であると仮定されるセキュリティ境界を示している。当該ディスプレイ/実行リポジトリ411は、デジタルワークへのアクセスに対して請求された料金を通知するためにクレジットサーバ414と、記憶されたデジタルワークにアクセスするためにリポジトリ415と、に更に連結されている。

【0024】使用権は、デジタルワークに直接アタッチされている。従って、デジタルワークの構造を理解することは重要である。特別な複合デジタルワークにおいては、デジタルワークの構造は、当然、階層のような非循環構造になるように編成されている。例えば、雑誌は、色々な人によって作成され所有されている可能性のある様々な記事や写真を載せている。これらの記事や写真の各々は階層構造におけるノードを示すことができる。結果的に、制御、即ち使用権は、クリエイタ（作成者）によって各ノードへ配置されてもよい。制御と料金の請求が各ノードに対応付けされるのを可能とすることによって、ワークのクリエイタは、権利及び料金が不正に欺かれないことを保証することが可能である。

【0025】本発明の実施の形態においては、デジタルワークに対するファイル情報は、「内容」ファイルと「記述ツリー」ファイルの二つのファイルへ分割される。リポジトリから見ると、「内容」ファイルは、そのフォーマットが、当該デジタルワークを再生し、ディスプレイし、又はプリントするために使用されるインタープリタに完全に依存しているアドレス可能なバイトの流れである。「記述ツリー」ファイルは、デジタルワークの内容とは無関係に、ワークに対する権利や料金を

検討することを可能とする。なお、本明細書中で使用されている用語「記述ツリー」は、デジタルワークの種々の構成要素同士の間の関係を示すために使用される任意のタイプの非循環構造を称する。

【0026】図5は、内容ファイルのレイアウトを示す。図5に関しては、デジタルワーク509は、ストーリーA:510、広告511、ストーリーB:512、及びストーリーC:513を備える。デジタルワークは相対アドレス0からスタートして記憶されると仮定される。デジタルワークの部分の各々は線形で記憶され、これによりストーリーA:510は、大体、0から30,000番目までのアドレスに記憶され、広告511は30,001から40,000番目までのアドレスに記憶され、ストーリーB:512は40,001から60,000番目までのアドレスに記憶され、ストーリーC:513は60,001から90,000番目までのアドレスに記憶される。ストーリーA:510の詳細は図6に示されている。図6に関しては、ストーリーA510はさらに小さく分けられて、大体、0から1500番目までのアドレスに記憶されたテキスト614、1501から10,000番目までのアドレスに記憶されたレンガ状に積まれた写真615、10,000から25,000番目までのアドレスに記憶されたグラフィックス616、25,001から30,000番目までのアドレスに記憶されたサイドバー617を示している。なお、内容ファイルにおけるデータは圧縮されるか(記憶保存のため)又は暗号化(セキュリティのため)されてもよい。

【0027】デジタルワークが階層としてのその構成要素部分によって示されることが容易に観察される。デジタルワークに対する記述ツリーは、関連した記述子ブロック(d-ブロック)のセットを備える。各d-ブロックの内容は、図7に関して記述されている。図7に関しては、d-ブロック700は、リポジトリ内のワークに対する唯一の識別子である識別子701と、当該ワークの第1バイトの開始アドレスを提供する開始アドレス702と、当該ワーク内のバイトの数を付与するレンジ703と、許諾された使用権及びそれらの状態データが保持されている権利部分704と、親のd-ブロックを指す親ポインタ705と、子のd-ブロックを指す子ポインタ706とを含んでいる。好ましい実施の形態において、識別子701は二つの部分を有している。第1の部分は、製造時にリポジトリへ割り当てられる唯一の数である。第2の部分は、作成時にワー

クへ割り当てられる唯一の数である。権利部分704は、権利に対応する種々の情報が保持されているルックアップテーブルのようなデータ構造を含む。それぞれの使用権によって要求された情報は、以下にもっと詳細に記述される。d-ブロックは厳格な階層を形成する。ワークのトップのd-ブロックは親を有しておらず、他の全てのd-ブロックは一つの親を有する。親と子のd-ブロック同士の間の使用権の関係と、競合の解消の方法が以下に説明されている。

10 【0028】特別なタイプのd-ブロックは「シェル」d-ブロックである。シェルd-ブロックはその部分の内容以外の新しい内容を追加しない。シェルd-ブロックは権利と料金の情報を追加するために一般的にデジタルワークのディストリビュータによって使用される。

【0029】図8は、図5のデジタルワークに対する記述ツリーを示す。図8に関しては、デジタルワークに対するトップのd-ブロック820は、内部に含まれた種々のストーリーと広告を指示する。この時点で、トップのd-ブロック820は、(ストーリーA:510を表す)d-ブロック821、(広告511を表す)d-ブロック822、(ストーリーB:512を表す)d-ブロック823、及び(ストーリーC:513を表す)d-ブロック824を指示する。

20 【0030】図9にはストーリーA:510に対する記述ツリーの部分が示されている。d-ブロック925はテキスト614を表し、d-ブロック926は写真615を表し、d-ブロック927はグラフィックス616を表し、d-ブロック928はサイドバー617を表す。

30 【0031】記述子ブロックの権利部分704は、図10に更に示されている。図10は、各権利に対して権利部分704において反復される構造を示している。図10に関しては、各権利は、権利コードフィールド1050及び状態情報フィールド1052を有している。権利コードフィールド1050は権利に割り当てられた唯一のコードを有する。状態情報フィールド1052は、権利の状態とデジタルワークに関する情報を含む。このような情報は以下の表1に示されている。権利部分704内に記憶される権利は、一般に、権利コードに基づく

40 数の順序であってもよい。

【0032】

【表1】

表 1

デジタルワーク状態情報

特性	値	使用
使用コピー数	枚数	使用中ワークのコピー数カウンタ。他のコピーが使用される時にインクリメントされ、使用が終了すると、デクリメントされる。
貸出し期間	タイムユニット	ドキュメントが貸出されるタイムユニットの最大数のインジケータ
貸出人のコピー	ブール代数值	現在のワークがオーソライズされたデジタルワークの貸出されたコピーであることのインジケータ。
残り時間	タイムユニット	課金されるドキュメント権利に対する使用残り時間を示すインジケータ。
ドキュメント記述	ストリング	ドキュメントについての種々の識別情報を含むストリング。これの正確なフォーマットは指定されていないが、発行者名、著者名、ISBN番号等の情報を含む
収入所有者	RO記述	デジタルワークに対する収入所有者を識別するハンドル。これは使用料を通知するために使用される。
発行日	日付記述	デジタルワークが出版された日。
ヒストリリスト	ヒストリ記録	デジタルワークをコピー、転送バックアップし、復元する動作のリポジトリ及び日付を記録する事象リスト。

【0033】記述データを内容から分離することによってデジタルワークを示すアプローチは、ファイルの各部分が同一の境界内に存在するが、内容の実際表示には関わっていないことを仮定している。特に、それは、内容表示がオブジェクト指向のアプローチを取るか否かの問題には中立である。内容をオブジェクトとして示すことは自然である。原理的に、請求構造を含む内容オブジェクトと、d-ブロック内で示される権利情報とを有することは都合が良い。表示のデザインにおいてこのようなバリエーションは可能であり、また代替的に実行可能であるが、例えば、オブジェクトの解説のような処理オーバーヘッドを導入することもある。

【0034】デジタルワークは階層ファイルシステムの一部としてリポジトリ内に記憶される。フォルダ（ディレクトリ及びサブディレクトリとも呼ばれる）は、デジタルワークと共に他のフォルダを含む。フォルダ内のデジタルワーク及びフォルダは、アルファベット順に並べられる。これらのデジタルワークはこれらのファイルがいかんして使用されるかを反映するようにタイプされる。フォルダそれ自体がデジタルワークとして取り扱われるように、使用権はフォルダにアタッチされ得る。フォルダへのアクセスは、あらゆる他のデジタルワークと同じ方法で処理される。以下により詳細に記述されるように、フォルダの内容はそれら自体の権利に

従属する。さらに、ファイルマネジメント権利がフォルダにアタッチされて、いかんしてフォルダ内容が管理され得るかを定義する。

- 30 【0035】使用権がデジタルワークの一部として取り扱われることは本発明に於ける基本である。デジタルワークが配給されると、許諾された使用権の範囲はそのまま変わらないか又は狭められる。例えば、デジタルワークがドキュメントサーバからリポジトリへ転送されると、使用権は、コピーを所定期間貸出す権利を含む（オリジナル権と呼ばれる）。リポジトリがデジタルワークのコピーを貸出す時、貸し手のコピーの使用権（次の権利のセットと呼ばれる）は、コピーを貸出すための任意の付加的権利を禁止するようにセットされることができ。基本的な考えは、彼らが持っている権利より以上の権利は許諾できないということである。

- 40 【0036】使用権のデジタルワークへのアタッチメントは種々の方法で行われることができる。デジタルワーク全体に対して使用権が同じである場合、デジタルワークがデジタルワークサーバにおけるデポジットのために処理される時、当該使用権が（デジタルワークへ）アタッチされる。種々の構成要素毎に異なる使用権を有するデジタルワークの場合、（使用権のアタッチは）デジタルワークの作成中に実行され得る。
- 50 使用権アタッチの自動化されたプロセスを提供する著作権ツ

ル又はデジタルワークアセンブリングツールが使用され得る。

【0037】以下に記述されるように、デジタルワークがコピーされ、転送され、又は貸出される時、「次の権利のセット」が指定され得る。この「次の権利のセット」はそれが伝送される時にデジタルワークへアタッチされる。

【0038】デジタルワークの各部分がそれ自体の使用権を有するかもしれないので、「含まれている部分」の権利がその親又はコンテナ(container)部分とは異なるインスタンスがある。これにより、何時、どのように権利が行使されるかを指定するために競合ルールが設定されなければならない。デジタルワークの階層ストラクチャ(構造)はこのようなルールの実施を円滑にする。「厳格な」ルールとは次のようになる: その部分と、その部分を含む先祖のd-ブロックと、全ての子孫のd-ブロックに対して許諾された場合、及びその場合だけにおいて、デジタルワークの当該部分に対する権利が許諾される。許諾されることにより、(1) それぞれの部分の各々が権利を有していなくてはならないことと、(2) その権利を行使するためのあらゆる条件が満たされることが示される。

【0039】もっと寛大なルールを用いて本発明を実行することも可能である。もっと寛大なルールにおいては、その部分へのアクセスが、権利を有する子孫の部分に対しては動作可能とされるが、権利を有さない子孫に対しては拒絶される。

【0040】厳格なルール及び寛大なルールの使用例が図11に関して示されている。図11に関しては、ルートd-ブロック1101は子d-ブロック1102乃至1105を有している。この場合、ルートd-ブロックは雑誌を示し、子d-ブロック1102乃至1105の各々は雑誌における記事を示す。厳格なルールに従ったルートd-ブロック1101によって示されたデジタルワークをPRINTするための要求が行われると仮定する。次に、このルートd-ブロック1101と子d-ブロック1102乃至1105に対する権利が検討される。ルートd-ブロック1101と子d-ブロック1102と1105のPRINT権は許諾される。子d-ブロック1103のPRINT権は許諾されないが、子d-ブロック1104は使用料の支払いを条件としてPRINT権を得る。

【0041】厳格なルール下では、子d-ブロックがPRINT権(単体)を有していないので、PRINT権は行使されることができない。寛大なルール下では、結果が異なる。子d-ブロック1102と1105によって示されたデジタルワークは、プリントされることができ、d-ブロック1104によって示されたデジタルワークは、使用料が支払われれば、プリントされることができ、d-ブロック1103によって示されるデジタル

ワークだけがプリントされることができない。要求が個別デジタルワークの各々に指定された場合、厳格なルール下では同じ結果が達成される。

【0042】本発明は、アクセスの許容及び非許容の様々なコンビネーションをサポートする。さらに、以下に説明されるように、使用権文法は、デジタルワークの所有者が、制約条件がコンテナ部分によってワークに課せられるか否かを指定するのを許容する。デジタルワークが使用権競合のために認可され得る方法は、実施に関する特性であり、この方法は、デジタルワークの性質に依存する。

【0043】図2の記述において、リポジトリが種々の形式を取ることが示されている。全てのリポジトリはデジタルワーク転送のためのサービスのコアセットを提供する。デジタルワークが交換される方法は、リポジトリ同士の間の全てのトランザクションに対する基本である。種々のリポジトリタイプはそれらが実行する最終的な機能が異なっている。リポジトリはそれら自体がデバイスであってもよいし、又は他のシステムに組み込まれてもよい。図2のレンダリングリポジトリ203がその一例である。

【0044】リポジトリはリポジトリ識別子と自身を対応付ける。一般的に、リポジトリ識別子は製造時に当該リポジトリへ割り当てられた唯一の数字である。各リポジトリはまた、特別なセキュリティクラス内にあるものとして分類される。いくつかの通信及びトランザクションは、特別なセキュリティクラス内にあるリポジトリ上で条件が付与される。種々のセキュリティクラスは以下に詳細に記述される。

【0045】動作に欠くことのできないものとして、リポジトリは識別証明書の所有を必要とする。識別証明書は偽造を防止するために暗号化され、Master repository (マスタリポジトリ) によって発行される。マスタリポジトリはリポジトリがデジタルワークを受け取るのを可能にするオーソライゼーションエージェントの役割を果たす。識別証明書は定期的に更新されなければならない。識別証明書は登録トランザクションに関して以下により詳細に説明される。

【0046】リポジトリはハードウェアと機能表現の両方を備えている。機能表現は、一般に、ハードウェアの具体化部分で実行されるソフトウェアである。或いは、機能表現は、Application Specific Integrated Circuit (ASIC) チップのようなハードウェアの具体化部分に埋め込まれてもよい。

【0047】リポジトリのハードウェアの具体化部分は、合意が得られれば、リポジトリを動作不可能にさせるセキュリティハウジングに囲まれている。リポジトリのハードウェアの具体化部分の基本的構成要素は図12に関して記述されている。図12に関しては、リポジトリは、処理手段1200、記憶システム1207、クロ

ック 1205 及び外部インタフェース 1206 からなる。処理手段 1200 はプロセッサエレメント 1201 とプロセッサメモリ 1202 を備える。当該処理手段 1200 は、コントローラと、リポジトリトランザクションと、このリポジトリのための使用権トランザクション機能とを提供する。デジタルワークとトランザクションメッセージの復号及び／又は圧縮解除のようなリポジトリの動作上の種々の機能も処理手段 1200 によって実行される。プロセッサエレメント 1201 は、マイクロプロセッサ又は他の適切な演算構成要素であってもよい。プロセッサメモリ 1202 は、一般的に、ROM（リードオンリーメモリ）及び RAM（ランダムアクセスメモリ）を更に備える。このようなメモリは、リポジトリの機能を実行する時に、プロセッサエレメント 1201 によって使用されるソフトウェア命令を含む。

【0048】記憶システム 1207 は、記述子記憶装置 1203 及び内容記憶装置 1204 を更に備える。記述子記憶装置 1203 は、デジタルワーク用の記述ツリーを記憶し、内容記憶装置 1204 は対応する内容を記憶する。記述子記憶装置 1203 及び内容記憶装置 1204 は、同じタイプの記憶媒体である必要はないし、また、必ずしも同じ物理的デバイス上に置かれる必要もない。ここで、例えば、記述子記憶装置 1203 は、（記述ツリー情報の迅速な検索のための）固体記憶装置上に記憶されてもよいが、内容記憶装置 1204 は光ディスクのような大容量記憶装置上に置かれてもよい。

【0049】クロック 1205 は、使用権のための又はデジタルワークに対応され得る使用料を課金するための種々のタイムベースの条件をタイムスタンプするために使用される。当該クロック 1205 は、タイムスタンプの完全性を維持するため、途絶しない電源、例えば、バッテリーを有する。外部インタフェース手段 1206 は、他のリポジトリ及びクレジットサーバへ信号接続を提供する。外部インタフェース手段 1206 は、RS-232 又は Personal Computer Manufacturers Card Industry (PCMCIA) 規格、又は FDDI のような標準インタフェースを介して信号の交換を提供する。外部インタフェース手段 1206 はネットワーク接続性を提供してもよい。

【0050】リポジトリの機能的な具体化部分は図 13 に記述されている。図 13 に関しては、機能的な具体化部分は、オペレーティングシステム 1301、コアリポジトリサービス 1302、使用トランザクションハンドラ 1303、リポジトリ特定機能 1304、及びユーザインタフェース 1305 を備える。オペレーティングシステム 1301 はリポジトリにとって特定であり、一般的に使用されているプロセッサのタイプに依存する。オペレーティングシステム 1301 は、リポジトリの基本的構成要素同士の間で制御し、インタフェースするための基本的サービスも提供する。

【0051】コアリポジトリサービス 1302 は、各リポジトリ及び全てのリポジトリによって要求される機能のセットを備える。このコアリポジトリサービス 1302 は、以下に詳細に定義されるセッション開始トランザクションを含む。このサービスのセットは、デジタルチケットを「パンチ」するために使用されるジェネリックチケットエージェントと、オーソライゼーション仕様を処理するためのジェネリックオーソライゼーションサーバとを含んでいる。デジタルチケット及びオーソライゼーションは、デジタルワークの配給及び使用を制御するための特定メカニズムであり、以下により詳細に説明されている。複数の識別証明書 1306 がコアリポジトリサービスに連結されていることに注目されたい。識別証明書 1306 はリポジトリの使用を可能とするために必要とされる。

【0052】使用トランザクションハンドラ 1303 は、デジタルワークへのアクセス要求を処理すると共にアクセスに基づいて料金を請求する機能を備えている。サポートされている使用トランザクションはリポジトリタイプにより異なる。例えば、デジタルワークに対するアクセス要求をハンドル（処理）することが必要ではないリポジトリもある。

【0053】リポジトリ特定機能 1304 はリポジトリにとって唯一の機能を備える。例えば、マスタリポジトリはデジタル的証明書を発行すると共に暗号化キーを保持するための特別な機能を有している。このリポジトリ特定機能 1304 は、リポジトリに対するユーザインタフェースの実行を含む。

【0054】幾つかのデジタルワークに対しては、オーソライズされていないコピーのあらゆる個々のインスタンスによって生じる損失は重要ではなく、主な経済的関心は、アクセスの利便性やロー・オーバーヘッドの請求を確実とすることにある。このような場合において、セキュリティの対策や保証が充分でなくても、単純かつ廉価なハンドヘルド・リポジトリやネットワークベースのワークステーションが適切リポジトリであるかもしれない。

【0055】封切られたばかりの映画、有価証券、又は株式証券のような幾つかのデジタルワークの他の極端なケースは、非常に価値が高いため、それらがコピーされたり、偽造されることがないのを確実にするために、警戒やかなり入念なセキュリティ（機密保護）対策を取ることが賢明である。このようなデジタルワークを保持するのに適したリポジトリは、物理的完全性を確実にするための、及び使用前のオーソライゼーションを検証するための念入りな対策を有することができる。

【0056】汎用プロトコルを配列することによって、全ての種類のリポジトリは原則として相互通信することができる。しかしながら、幾つかのワーク（著作物）のクリエイタは、彼らのワークがセキュリティレベルが充

分に高いリポジトリへのみ転送を指定したいと望む。このため、ドキュメントリポジトリはセキュリティのクラス及びレベルに対してランキングシステムを有する。本

* 発明の好ましい実施の形態におけるセキュリティクラスは表 2 に説明されている。

* 【0057】

表 2
リポジトリセキュリティレベル

レベル	セキュリティ記述
0	オープンシステム。ドキュメント転送は暗号化されない。識別用デジタル証明書は要求されない。セキュリティ対策を欺く為には大した知識は必要とされないの、このシステムのセキュリティは殆どユーザの良心に委ねられる。リポジトリはオーソライズされていないプログラムのファイルが実行、アクセス又はコピーされるのを防止するために何も提供しない。システムは取り外し可能記憶装置の使用を防止しないし、記憶されたファイルを暗号化しない。
1	最低のセキュリティ。記憶されたファイルが最少限で暗号化されること除き、前のクラスと同様。
2	基本的なセキュリティ。プログラミング、リポジトリの内容又はクロック状態を処理するための特別なツール及び知識が要求されることを除き前のクラスと同様。全てのデジタル通信が暗号化される。デジタル証明書が識別として提供される。中間レベルの暗号化が使用される。リポジトリ証明書番号は偽造不可能。
3	一般的なセキュリティ。前のクラスと同様だが、リポジトリの物理的完全性を処理するために特別ツールが必要とされる。十分な暗号化が全ての転送に使用される。パスワード保護がローカルユーザインタフェースを使用するために必要とされる。デジタルクロックシステムはオーソライゼーションなしではリセットされない。ワークは取外し可能な記憶装置上では記憶されない。ワークをプログラムとして実行する時、このシステムはワークを独自のアドレススペース内でランし、システムコード又はワークを含む任意のファイル記憶装置又は他のメモリへの直接アクセスを提供しない。ワークへは転送トランザクションプロトコルを介してのみアクセスされる。
4	高レベルの暗号化が全ての通信上で使用されることを除き、前のクラスと同様。物理的及び電子的変更要求を記録するためにセンサーが使用される。このような変更の後、リポジトリは指定されたサーバへ変更を通知する迄は他のトランザクションを実行しない。
5	物理的又はデジタルの変更要求が、リポジトリの物理的完全性、又はデジタル及び暗号バリアの完全性を脅かす事前設定された閾値を越えた場合、リポジトリは履歴のドキュメント記述記録のみ保存するが、不謹慎な相手にリリースされると悪用されそうなデジタル識別を消去又は破壊する。このことを除いては前のクラスと同様。物理的システムが処理されたことを示すために認証の証明書も修正する。指定されたドキュメントの内容も消去する。

6	リポジトリが、変更を通知するためにワイヤレス通信を行い、ノイズな警報を用いることを除き、前のクラスと同様である。
10	これは非常に高いレベルのセキュリティに相当する。このサーバは、トランザクション、センサ読み取り、及びセキュリティを欺こうとする要求を通知する遠隔セキュリティシステムへ一定の通信を維持する。

【0058】表2に記述されたセキュリティレベルの特性表示は固定的なものではない。もっと重要なのは、異なるリポジトリに対して異なるセキュリティレベルを有することの考えである。新しいセキュリティクラス及び必要条件が社会的状況やテクノロジーの変化に応じて進化することが望まれる。

【0059】リポジトリ・ユーザ・インタフェース

ユーザインタフェースは、ユーザがリポジトリと対話してデジタルワークへのアクセスを得るためのトランザクションを呼出すか又は使用権を行使するメカニズムとして広く定義付けられている。上記のように、リポジトリは種々の形式で組み込まれる。リポジトリに対するユーザインタフェースは、特定の実施例によって異なる。ユーザインタフェースは、デジタルワークを示すアイコンや実行され得る種々のトランザクションを有するグラフィカルユーザインタフェース（GUI）であってもよい。ユーザインタフェースは情報がユーザにプロンプトされる生成ダイアログであってもよい。

【0060】ユーザインタフェース自体はリポジトリの一部である必要はない。リポジトリが他のデバイス内に埋め込まれてもよい場合、ユーザインタフェースは、リポジトリが埋め込まれるデバイスの一部であるにすぎない。例えば、リポジトリはコンピュータシステム内の利用可能なスロットに挿入される「カード」内に埋め込まれることができる。ユーザインタフェースは、コンピュータシステムで実行されるディスプレイ、キーボード、カーソル制御デバイス及びソフトウェアの組み合わせである。

【0061】最低でも、ユーザインタフェースは、ユーザが、アクセス要求及び英数字データのような情報を入力すると共にトランザクション状態に対してフィードバックを提供するのを可能にしなければならない。ユーザインタフェースは、次に、要求を保守するために、リポジトリに対して、適切なトランザクションを開始させる。特別なユーザインタフェースの他のファセットはリポジトリが提供する機能に依存する。

【0062】本発明において、料金は権利の行使に対応付けられる。料金支払の必要条件是、使用権言語における使用権の各バージョンと共に記述される。このような料金の記録及び通知はクレジットサーバによって行われる。料金を権利に対応付けることによって可能とされるいくつかの可能性の内の一つは、広範囲の請求モデルを

サポートすることの可能性である。従来のソフトウェアによって使用された最も簡単なモデルは、購入時には単一料金しか設定されておらず、その後は、購入者は彼女は彼女が望む限り何度もワークを使用できる無制限の権利を得たのである。他のモデルとしては、課金される使用及び可変料金を含む。単一ワークは異なる使用に対して異なる料金を有することができる。例えば、ディスプレイ上の写真を見ることは、ハードコピーを作成すること又は当該ハードコピーを新しく作成されたワークに組み込むことは別の料金を有する。これらの他の請求モデルへのキーは、料金を設定してこれらのトランザクションに対する料金計算を行うロー・オーバーヘッド手段を有することにある。

【0063】クレジットサーバは、料金が請求され支払われるよう、これらのトランザクションを確実にオーソライズし、記録する計算システムである。当該クレジットサーバは料金を請求クリアリングハウスへ通知する。当該請求クリアリングハウスは、金融トランザクションが発生するとこれらのトランザクションを管理する。結果的には、請求書が発生され、その料金が清算される。好ましくは、クレジットサーバは、料金トランザクションを記憶し、清算のためにネットワークを介して請求クリアリングハウスと周期的に通信する。このような実施例において、請求クリアリングハウスとの通信は、完全性とセキュリティの理由のために暗号化される。他の実施例において、クレジットサーバは、ユーザの勘定に対してトランザクションが「リアルタイム」で発生する「デビット（借り方）カード」として作用する。

【0064】クレジットサーバは、メモリ、処理手段、クロック、並びにリポジトリ及び金融機関と結合するためのインタフェース手段（例えば、モデム）からなる。クレジットサーバは、セキュリティ及び認証の機能を有する必要もある。これらのエレメントはリポジトリのエレメントと本質的に同じエレメントである。従って、単一デバイスが対応する機能とプロトコルを実行するための適切な処理エレメントを有していると仮定すれば、それはリポジトリとクレジットサーバの両方であることができる。一般的に、しかしながら、クレジットサーバは、リポジトリに結合され、以下に説明されるように金融トランザクションを介して対話する。金融機関との対話は、金融機関自体によって設定されたプロトコルを介して行われてもよい

10

20

30

40

50

【0065】本発明の実施の形態において、サーバとリポジトリの両方に対応しているクレジットサーバは、請求クリアリングハウス（情報の中心機関）へ金融トランザクションを通知する。例えば、デジタルワークがその料金に関して一つのリポジトリから他のリポジトリへコピーされる時、リポジトリの各々に結合されたクレジットサーバはこのトランザクションを請求クリアリングハウスに通知する。このことは、クレジットサーバと請求クリアリングハウスの通信が中断された場合でも、トランザクションの料金計算が行われることを保証する観点から望ましい。しかしながら、幾つかの実施例では、いくつかのトランザクションを失う危険性のある時にトランザクションの処理を最小とするように当該トランザクションへ通知する単一クレジットサーバのみを組み込むことができる。

【0066】本発明は、デジタルワークとそれらの部分に対応している権利を定義するために高レベル「使用権言語」のステートメントを用いる。使用権ステートメントはリポジトリによって解釈され、あるデジタルワークに対してどのトランザクションが無事に遂行されるかを決定するため、及びこれらのトランザクションに対するパラメータを決定するために使用される。例えば、言語におけるセンテンスは、所与のデジタルワークがコピーされ得るか否か、何時そしていかにしてそれが使用され得るか、及びその使用に対していくら（あれば）請求されるべきかを決定する。使用権ステートメントが発生すると、これらは、トランザクションの処理中にアクセスするために適切な形式でコード化される。

【0067】使用権をデジタルワークの階層表現と組み合わせることで言語で定義することによって、（デジタルワークの）広範囲の配給と料金方式のサポートが可能となる。一つの例は、権利の複数のバージョンをワークにアタッチする可能性である。そこで、クリエイタは10ドルで5部のコピーを作成するPRINT 権や、100ドルでコピーを無制限に作成するPRINT 権をアタッチすることができる。次に、購入者は、どのオプションが彼のニーズに一番適しているかを選択することができる。他の例としては、権利と料金が加算されるものである。ここで、複合ワークの場合、構成要素ワークの各々の権利及び料金は、全体としてワークに対する権利及び料金を決定するときに使用される。

【0068】権利の基本的な内容は、図14に示されている。図14に関しては、権利1450は、トランザクショナル構成要素1451及び仕様構成要素1452を有する。権利1450は、権利によって組み込まれる使用又は配給の特権を示すラベル（例えば、COPY又はPRINT）を有する。トランザクショナル構成要素1451は、デジタルワークが使用されるか又は配給される特別な方法に対応している。当該トランザクショナル構成要素1451は、一般的に、権利についての使用又は配

給の特権を実行するリポジトリ内のソフトウェア命令の中に組み込まれている。仕様構成要素1452は、権利が行われる前に満たされなければならない条件を指定するか又は種々のトランザクションに関連したパラメータを指定するために使用される。本発明の好ましい実施の形態においては、これらの仕様は、コピーカウント1453、料金と報酬1454、タイム1455、アクセスとセキュリティ1456、及び制御1457を含む。これらの仕様の各々は、言語文法エレメントに関して以下に詳細に説明されている。

【0069】使用権言語は以下に説明される文法に基づく。文法は、言語に対するシンボルの有効なシーケンスを定義するために便利な手段である。文法を説明する際、代替物からの明確な選択を示すために表記 $[a | b | c]$ が使用される。この例において、センテンスは、 $"a"$ 、 $"b"$ 、又は $"c"$ のいずれかを有することができる。センテンスはその内の一つを正確に含んでいなければならない。ブレース $\{\}$ が選択項目を示すために使用される。ブラケット $[\]$ やバーやブレースが使用権センテンスの言語を記述するために使用されるが、言語の実際のセンテンス内には現れない。

【0070】これに対して、括弧 $()$ は使用権言語の部分である。括弧はリストされた項目をグループ化するために使用される。表記 (x^*) は可変長リスト、即ち、タイプ x の一つ以上の項目を含むリストを示すために使用される。表記 $(x)^*$ は x を含む可変数リストを示すために使用される。

【0071】文法におけるキーワードはコロンが付加されるワードである。キーワードは言語における共通で、非常に特別なケースである。これらは単一値、一般には識別子を示すために時々使用される。多くの場合、キーワードとパラメータは全体的にオプションである。キーワードが与えられると、その値として単一の識別子を取る。いくつかの場合において、キーワードは識別子のリストを取る。

【0072】使用権言語において、時間は、hours:minutes:seconds（又はhh:mm:ss）表現で指定される。タイムゾーンインジケータ、例えば、PDT (Pacific Daylight Time) も指定されることができる。日付は年/月/日（又はYYYY/MM/DD）として示される。これらの時間及び日付の表現は、時刻即ち時間の単位を指定することができる。お金の単位はドルに換算して指定される。

【0073】使用権言語において、種々の「もの」が互いに対話する必要がある。例えば、使用権のインスタンスは、銀行トランザクション、デジタルチケット等を指定する。このような物は、識別される必要があり、本発明においては接尾辞 $^{-ID}$ を用いて指定される。

【0074】使用権文法は図15にその全体のリストが作成されており、以下に詳細に記述されている。

【0075】文法エレメント1501 "Digital Work Rights=(Rights*)" は、デジタルワーク権利を権利のセットとして定義する。デジタルワークにアタッチされる権利のセットはそのデジタルワークがいかにして転送され、使用され、実行され又は再生されるかを定義する。権利のセットは全体的なデジタルワークにアタッチされ、複合デジタルワークの場合はデジタルワークの構成要素の各々にアタッチされる。デジタル構成要素の使用権は異なる。

【0076】文法エレメント1502 "Right:=(Right-Code [Copy-Count] [Control-Spec] [Time-Spec] [Access-Spec] [Fee-Spec])" は、権利の内容を列挙している。各使用権は権利コードを指定しなければならない。各使用権は権利が行使され得る前に満たされなければならない条件も選択的に指定することができる。これらの条件とは、コピーカウント、制御、タイム、アクセス、及び料金の条件である。本発明の好ましい実施の形態において、オプションエレメントに関しては、以下のデフォルトが適用される。即ち、1のコピーカウント、権利の使用に関する時間制限の非設定、権利を使用するために要求されるアクセステスト又はセキュリティレベルの非設定、及び無料。これらの条件はそれぞれ以下により詳細に記述される。

【0077】それぞれが同じ権利コードを有する複数の権利バージョンをデジタルワークが有していてもよいことに注目することは重要である。複数のバージョンは、デジタルワークにアクセスするための代替条件や料金を提供する。

【0078】文法エレメント1503 "Right-Code:=Render-Code | Transport-Code | File-Management-Code | Derivative-Works-Code Configuration-Code" は、(各権利が異なる権利コードによって識別されるが、) 特定権利の各々を特別な権利タイプに区別する。このように、文法は、デジタルワークの部分に対応し得る可能性のある権利のカテゴリを提供する。以下において、権利はそれらを記述する便宜上、カテゴリ別に分けられる。

【0079】文法エレメント1504 "Render-Code:=Play: {Player:Player-ID} | Print: {Printer:Printer-ID}" は、全てがデジタルワークの一時的で、短命の、又はノンデジタルのコピーの作成を含む権利のカテゴリのリストである。これらのコピーは使用された後は消去される。

・Play- 幾つかのプロセッサ上でデジタルワークをレンダリング又は実行するプロセス。これはデジタル映画の再生、デジタル音楽の再生、ビデオゲームの再生、コンピュータプログラムの実行、又はディスプレイ上のドキュメントの再生などのものを含む。

・Print- 使用権によってこれ以上は保護されない媒体へワークをレンダリングすること、例えば、用紙にプリ

ントすること。

【0080】文法エレメント1505 "Transport-Code:=Copy | Transfer | Loan {Remaining-Rights:Next-Set-of-Rights} [(Next-Copy-Rights:Next-Set of Rights)]" は、持続性があり、他のリポジトリ上で使用可能なデジタルワークのコピーの作成を含む権利のカテゴリのリストである。オプションなNext-Copy-Rightsは、伝送された後のワークの権利を決定する。これが指定されない場合、伝送されたコピーの権利はオリジナルの権利と同じである。オプションなRemaining-Rightsは、コピーが貸出された時にデジタルワークと共に残っている権利を指定する。この権利が指定されない場合、デフォルトは、コピーが貸し出された時に権利が行使されないデフォルトである。

・Copy- ワークの新しいコピーを作成する。

・Transfer- ワークを一つのリポジトリから他のリポジトリへ移動する。

・Loan- 指定された期間、コピーを他のリポジトリへ貸し出す。

20 【0081】文法エレメント1506 "File-Management-Code:=Backup {Back-Up-Copy-Rights:Next-Set-of-Rights} | Restore | Delete | Folder | Directory {Name:Hide-Local | Hide-Remote} [Parts:Hide-Local | Hide-Remote]" は、コピー所有者を悲惨な機器の故障から保護するためにバックアップコピーのようなファイルマネージメントのための動作を含む権利のカテゴリのリストである。

【0082】多数のソフトウェアライセンス及び著作権法は、コピー所有者を悲惨な機器の故障から保護するためにバックアップコピーを作成する権利をコピーの所有者に与える。しかしながら、制御されないバックアップコピーの作成は、オーソライズされたコピーが販売された後でも制御されないバックアップコピーが保存され復元されることもできるので、本質的に使用を制御する能力とならない。

【0083】File-Management (ファイルマネージメント) の権利は、使用権を尊重する方法でバックアップコピーの作成及び復元を可能とし、コピー所有者と権利授与者及び収入所有者の双方の要求条件を尊重する。ワーク記述のバックアップコピー (使用権及び料金データを含む) は、適切なプロトコル及び使用権制御下において、十分に高度なセキュリティを有する他のドキュメントリポジトリへ送られ得る。更なる権利によってデジタルワークのフォルダへの編成が許容される。これらのフォルダはこれら自体がデジタルワークとして扱われ、それらの内容が、リポジトリの内容を決定しようとするパーティから「隠蔽」されてもよい。

・Backup- 媒体の故障に対する保護としてデジタルワークのバックアップコピーを作成すること。

50 ・Restore- デジタルワークのバックアップコピーを

復元すること。

・Delete—デジタルワークのコピーを削除又は消去すること。

・Folder—フォルダを作成及びネーミングし、フォルダ同士の間でファイル及びフォルダを移動すること。

・Directory—フォルダ又はその内容を隠蔽すること。

【0084】文法エレメント1507“Derivative-Work s-Code:= Extract | Embed | Edit {Process:Process-ID} {Next-Copy-Rights:Next-Set-of-Rights}”は、新しいワークを作成するためのデジタルワークの使用を含む権利のカテゴリのリストである。

・Extract—新しいワークの作成目的のため、ワークの部分を取り除くこと。

・Embed—既存のワーク内にワークを組み込むこと。

・Edit—既存のデジタルワークの部分のコピー、選択、及び変更することによりデジタルワークを変更すること。

【0085】文法エレメント1508“Configuration-Code:=Install | Uninstall”は、リポジトリ（一般的にレンダリングリポジトリ）上でソフトウェアをインストールする権利及びインストールしない権利のカテゴリのリストである。これは一般的にレンダリングリポジトリ内の新しいタイプの再生手段のインストールに対して発生する。

・Install—リポジトリ上で新しいソフトウェアをインストールすること。

・Uninstall—リポジトリから既存のソフトウェアを除去すること。

【0086】文法エレメント1509“Next-Set-Of-Rights:= { Add: Set-Of-Rights} | {(Delete:Set-Of-Rights) | {(Replace:Set-Of-Rights) | {(Keep:Set-Of-Rights)}}”は、デジタルワークのコピーに対する権利がどのように進められるかを定義する。Next-Copy-Rightsが指定されない場合、次のコピーに対する権利は現在コピーの権利と同じである。さもなければ、次のコピーに対する権利のセットが指定され得る。Addの後の権利のバージョンは、権利の現在セットに加算される。Deleteの後の権利は、権利の現在セットからデリートされる。権利コードだけがDeleteの後にリストされた場合、これらのコードを有する権利の全てのバージョンがデリートされる。Replaceの後の権利のバージョンは、権利の現在セットにおける同じタイプの権利の全てのバージョンを包含する。

【0087】Remaining-Rightsが指定されない場合は、全てのローンコピーが貸し出された後、オリジナルに対する権利はない。Remaining-Rightsが指定された場合、背後でどの権利をキープすべきかについての表現を単純化するためにKeep: トークンが使用されることができ。キープ手段に続く権利コードのリストは、リストされた権利のバージョンの全てが残りのコピーの中に保持

されていることを意味する。この仕様は、引き続きDelete: 又はReplace:仕様によって無効にされることができ。る。

【0088】種々のトランザクションに関しては、権利に対して同時に行使され得るワークの「コピー」の数について何らかの制限を行うことが所望される。例えば、一度に貸し出され、一度に閲覧され得るデジタルワークのコピー数を制限することが所望される。

【0089】文法エレメント1510“Copy-Count:= (Copies:positive-integer | 0 | unlimited)”は、権利を得ることを条件とするワークの「コピー」数を定義する条件を提供する。コピーカウントはゼロ、固定数、又は無制限である。デジタルワークに対して単一のコピーカウントが存在していることに対抗して、コピーカウントは各権利に対応している。権利に対するCopy-Countは権利が行使される度にデクリメントされる。権利に対するCopy-Countがゼロに等しい場合、この権利はもう行使されない。Copy-Countが指定されない場合、デフォルト値は1である。

20 【0090】権利と料金は、概して、クリエイタによって許諾された権利に依存すると共に、新しいディストリビュータによって課される更なる制限にも左右される。制御仕様は、クリエイタと、更なる制限及び料金の賦課を支配する彼らのディストリビュータと、の間での対話を処理する。例えば、デジタルワークのディストリビュータは、デジタルワークの最終的な消費者が、購入されたデジタルワークを商業的に活用して料金をつり上げたり、さもなければ、利益を得たりすることを望まない。

30 【0091】文法エレメント1511“Control-Spec:= (Control: {Restrictable | Unrestrictable} {Unchargeable | Chargeable})”は、権利の行使に対する親の使用権及び料金の影響を指定するための条件を提供する。より高いレベルのd-ブロックが、権利に対して更なる制限（タイム仕様及びアクセス仕様）を課することができる場合、デジタルワークは制限可能である。更なる制約が課せられない場合は制限不可能である。デフォルトの設定は制限可能である。それ以上の料金を権利の使用に課することができない場合、権利は請求不可能である。

40 50 【0092】何時権利が行使されるかの開始日を割当てたり又は期間を指定することがしばしば望まれる。文法エレメント1512“Time-Spec:= ({Fixed-Interval | Sliding-Interval | Meter-Time} Until:Expiration-Date)”は、権利の行使への時間の条件を指定することを備える。権利は指定された時間の間、許諾され得る。異なる時間仕様の種類は異なる種類の権利に対して適正である。幾つかの権利は決められた所定の期間、行使されこ

とができる。幾つかの権利は、あるトランザクションにより権利が呼び出される時刻からスタートするインターバルの間だけ行使されてもよい。幾つかの権利は、ある種類の課金時間に従って行使されるか又は請求されることができ、この課金時間は、分離されたインターバルへ分割され得る。例えば、絵画を1時間見る権利は、10分の閲覧を6回、15分の閲覧を4回、及び20分の閲覧を3回に分割されてもよい。

【0093】用語「タイム」及び「デート」は同義語として使用されて時刻を表している。いくつかの種類の時間仕様がある。各仕様は、使用権が適用される時間に対して制限を示す。Expiration-Date は使用権が終了する時刻を指定する。例えば、Expiration-Date が「1995年1月1日」である場合、権利は1995年の最初の時刻に終了する。Expiration-Date が「forever」と指定された場合、権利は終了せずに継続するものとして解釈される。満期日だけが与えられている場合、その権利はその満期日まで所望されるだけ行使されることができ

【0094】文法エレメント1513 Fixed-Interval:= From:Start-Time は開始時間から満期日までランされる所定のインターバルを定義するために使用される。

【0095】文法エレメント1514 Sliding-Interval:= Interval:Use-Duration は、不定の（又は「オープン」）なスタート時間を定義するために使用される。該エレメントは、内容がアクセス可能である時間の連続期間に制限をセットする。この期間は、最初のアクセスの時にスタートし、この期間が経過した後か又は満期日に達した後かのいずれか早い方で終了する。例えば、権利が10時間の連続アクセスを付与する場合、使用期間は、最初のアクセスが行われた時に開始し、10時間後に終了する。

【0096】文法エレメント1515 Meter-Time:=Time-Remaining:Remaining-Use が「課金時間」、即ち、権利が実際に行使された時間の測定、を定義するために使用される。これは、デジタルワーク使用中の時間（期間）が連続的である必要がない点で、Sliding-Interval仕様とは異なる。例えば、権利が三日間のアクセスを保証する場合、これらの日数は、一ヶ月以上にわたり分散されることができ。この仕様によって、これらの権利は、課金時間を使い果たすか又は満期日に達するか早い方まで行使される。

【0097】Remaining-Use:=Time-Unit

Start-Time:=Time-Unit

Use-Duration:=Time-Unit

時間仕様の全てはそれらの最終的なインスタンス生成においてタイムユニット仕様を含む。

【0098】本発明は、配給又は使用方式へ導入される種々のセキュリティメカニズムを提供する。文法エレメント1516 Access-Spec:=([SC:Security-Class]

[Authorization:Authorization-ID*] [Other-Authorization:Authorization-ID*] [Ticket:Ticket-ID]) はアクセス及び転送を限定する手段を提供する。アクセス仕様は権利を行使するためにリポジトリに対して要求されたセキュリティクラス又は満たされなければならない要求されたオーソライゼーションテストを指定することができる。

【0099】キーワード“SC:” はアクセスにおいて含まれるリポジトリに対する最小のセキュリティレベルを指定するために使用される。“SC:” が指定されない場合、最低のセキュリティレベルが受容可能である。

【0100】オプションな“Authorization” キーワードは、ワークと同じリポジトリでの要求されるオーソライゼーションを指定するために使用される。オプションな“Other-Authorization” キーワードは、トランザクションにおける他のリポジトリへ要求されるオーソライゼーションを指定するために使用される。

【0101】オプションな“Ticket:” キーワードはトランザクションに要求されたチケットの識別を指定する。デジタルチケットを含むトランザクションは、トランザクションが進められる前に「パンチ」することができるか又はこのチケットを有効にすることができる適切なデジタルチケットエージェントを検索しなければならない。チケットは以下により詳細に記述されている。

【0102】リポジトリとドキュメントサーバを含むトランザクションにおいて、いくつかの使用権は、リポジトリが特定のオーソライゼーションを有すること、サーバがいくつかのオーソライゼーションを有すること、又は両リポジトリが（恐らく異なる）オーソライゼーションを有することを要求することができる。オーソライゼーション自体は、他のデジタルワークと同様にリポジトリ同士の間で移動することができるデジタルワーク（これ以降、オーソライゼーションオブジェクトと呼ばれる）である。これらのコピー及び転送は、他のデジタルワークと同じオーソライゼーション及び料金を受けることができる。リポジトリはこのオーソライゼーションオブジェクトがこのリポジトリ内に含まれる場合、オーソライゼーションを有すると言われる。

【0103】いくつかのケースにおいて、オーソライゼーションはドキュメントサーバ及びリポジトリ以外のソースから要求されてもよい。Authorization-IDによって参照されるオーソライゼーションオブジェクトは、リポジトリとオーソライゼーションソースの間での通信リンクをセットアップするために使用されるデジタルアドレス情報を含むことができる。これらは電話番号に類似している。このようなアクセステストに対しては、通信が設定される必要があり、権利が行使される前にオーソライゼーションを得なければならない。

【0104】一度だけの使用権に関するこの方式の変形

は、デジタルチケットを持つことである。チケットはデジタルチケットエージェントに提供され、チケットにそのタイプが指定される。最も単純なケースにおいては、全てのリポジトリへ利用可能な、保証されたジェネリックチケットエージェントが、このチケットを「パンチ」するために利用可能である。他のケースにおいては、チケットは「特別」のチケットエージェントを検索するためのアドレス情報を含むことができる。チケットは一旦パンチされると、同じ種類のトランザクションに再度使用することはできない（以下に説明されるように、非パンチ状態にされる、即ちリフレッシュされる場合は除く）。パンチングは、チケットが使用される日時のタイムスタンプでチケットにマーキングすることを含む。チケットはデジタルワークであり、これらのデジタルワークの使用権に従って、リポジトリ同士の間でコピー又は転送され得る。

【0105】本発明の好ましい実施の形態において、「パンチされた」チケットは、それがコピーされるか又は抽出される時、「非パンチ状態にされる」即ち「リフレッシュされる」。Copy及びExtract 動作はそのデジタルチケットの特性として日時を保存する。チケットエージェントがチケットを付与された時、当該デジタルエージェントは、それが、パンチされた最終時間の後でコピーされたか否かを単にチェックすることができる。デジタルチケットは、当然、そのコピーを有するか、又は当該コピーにアタッチされた使用権を抽出しなければならない。

【0106】チケットをパンチしない能力は以下の場合に重要である。

- ・デジタルワークが一回だけの使用に限定されて低コストで頒布される。

- ・デジタルワークが、他のワークの購入の際の割引をするために一回使用可能なチケットによって頒布される。

- ・デジタルワークが、未来のアップグレード用を使用することができる（購入価格に含まれ、恐らくそのワーク内に埋め込まれた）チケットによって頒布される。

【0107】これらの各ケースにおいて、購入されたコピーが、（チケットを含む）デジタルワークから成される場合、新しい所有者は、そのコピーの販売者がそのワークを使用してもしなくても、フレッシュ（パンチされてない）チケットを得ることを期待する。これに対して、ワークの貸出し又はワークを単に他のリポジトリへ転送することは、チケットを再利用することではない。

【0108】デジタルワークの使用に対する請求は、商業的流通システムに対する基本である。文法エレメント1517 "Fee-Spec:= [Scheduled-Discount] Regular-Fee-Spec | Scheduled-Fee-Spec | Markup-Spec" はデジタルワークの使用に対する請求のオプションの範囲を提供する。

【0109】この方法の主要な特徴は潜在的に少量のトランザクションに対するローオーバーヘッド請求の開発である。これにより、数千のトランザクションの各々に対してほんの数セントの料金を集めることが容易となる。

【0110】文法は、使用毎に請求が発生する使用と、タイムユニットによって課金される使用と、を区別する。トランザクションは、デジタルワークを使用するためにユーザが支払う料金だけでなく、ユーザを誘導してデジタルワークを使用又は配給させるために権利許諾者がユーザに支払った報酬をサポートすることができる。

【0111】オプションなスケジュールされた割引は、料金仕様の残りの部分に対応し、所定割合の時間だけ料金を割り引く。指定されない場合、スケジュールされた割引はない。規定の料金仕様は時間当たりで一定している。スケジュールされた料金仕様は、料金仕様が変化する日付のスケジュールを付与する。マークアップ仕様は既に請求されている料金に対して手数料を加算するためにd-ブロック内で使用される。

【0112】文法エレメント1518 "Scheduled-Discount:=(Scheduled-Discount:(Time-Spec Percentage)

*)"。Scheduled-Discountは、本質的に、デジタルワークの権利の当該バージョンに対する他の料金仕様のスケジュールされた変更子である。（これは子又は親のデジタルワーク又は他の権利バージョンに対応しない）。これは時間とパーセンテージのペアリストである。トランザクションの時にまだ経過していないリスト内の最新時間が、実施される時間である。このパーセンテージは割引パーセントを付与する。例えば、数字10は10%の割引を示す。

【0113】文法エレメント1519 "Regular-Fee-Spec:=([Fee: | Incentive: | Per-Use-Spec | Metered-Rate-Spec | Best-Price-Spec | Call-For-Price-Spec {Min:Money-Unit Per:Time-Spec} {Max:Money-Unit Per:Time-Spec} To:Account-ID)" は数種類の料金仕様を提供している。

【0114】Fee:が指定されたならば、料金はコピー所有者/ユーザによって総収入所有者へ支払われる。Incentive:が指定されると、報酬は収入所有者によってユーザへ支払われる。Min:仕様が付与されると、その使用に対してタイム仕様毎に請求される最小の料金が示される。次に、Max:仕様が与えられると、その使用に対してタイム仕様毎に請求される最大の料金が示される。Fee:が指定されると、Account-IDは料金が支払われる勘定書を識別する。Incentive:が指定されると、Account-IDは料金が支払われる勘定書を識別する。

【0115】文法エレメント1520 "Per-Use-Spec:= Per-Use:Money-Unit" は、トランザクションに掛かった時間に関係なく、権利が行使される度に支払われる単純

な料金を定義する。

【0116】文法エレメント1521 "Metered-Rate-Spec:= Metered:Money-Unit Per:Time-Spec"は権利が行使された時間に応じて支払われる課金レート料金を定義する。このように、トランザクションを終了するのに掛かった時間がこの料金を決定する。

【0117】文法エレメント1522 "Best-Price-Spec:= Best-Price:Money-unit Max:Money-unit"は勘定書が設定された時に決定される最適価格を指定するために使用される。この仕様は、リポジトリが利用できない情報に依存する特別な取引(deal)、リベート、及び価格表示を含む。全ての料金仕様は、消費者が問屋であるか若しくは好ましい顧客であるか、又は販売者が何らかの方法でオーソライズされていることを示すことができるチケット又はオーソライゼーションと合成されることができる。Max: フィールドにおける金額は使用にかかる経費の最大の額である。これは、クレジットサーバに仮請求された金額である。しかしながら、最終的に取引が成立した時にいかなる超過金額も別のトランザクションで消費者へ返却される。

【0118】文法エレメント1523 "Call-For-Price-Spec:= Call-For-Price"は、価格が変動する場合を含むことを意図している点で、"Best-Price-Spec" に類似している。Call-For-Price-Spec は、価格を決定するためにディーラーとの情報交換を要求する。権利が行使される時にリポジトリがディーラーと通信できない場合、このオプションは行使されることができない。このことは、ディーラーが、権利を行使するためにプライスを付け、取引証明を渡す安全なトランザクションに基づいており、この取引証明は請求プロセスにおいて参照されたり又は取込まれる。

【0119】文法エレメント1524 "Scheduled-Fee-Spec:=(Schedule:(Time-Spec Regular-Fee-Spec) *)" は、料金仕様の変化する日付のスケジュールを提供するために使用される。最新の過去の日付を有するこの料金仕様が、実施される料金仕様である。これは料金の合意を各期間ごとに変更するための手段を提供するので、より一般的である。

【0120】文法エレメント1525 "Markup-Spec:=Markup:percentage To: Account-ID"は、既に請求されている料金に手数料を加算するために提供される。例えば、5%のマークアップはこれまでの累積料金の5%の料金がディストリビュータに割り当てられることを示す。マークアップ仕様は他の種類の料金仕様の全てに適用されることができる。当該マークアップ仕様はディストリビュータによって提供されたシェル内に一般に使用される。これは現在d-ブロックの部分であるd-ブロックに対応する料金に当てはまる。これは税金又はディストリビュータオーバーヘッドに使用するために役立つ仕様である。

【0121】ユーザがデジタルワークへのアクセスを要求する時、リポジトリは種々のトランザクションを開始する。呼び出されたトランザクションの組み合わせは使用権に割り当てられた仕様に依存する。セッション開始トランザクション、金融トランザクション及び使用トランザクションの三つの基本的なタイプのトランザクションがある。概して、セッション開始トランザクションは有効なセッションを設立するために最初に開始される。有効なセッションが設定されると、種々の使用権に対応するトランザクションが呼び出される。最後に、特定トランザクションリクエスト(request specific transactions) が実行される。

【0122】トランザクションは、二つのリポジトリ(一つがサーバとして作用する)の間、リポジトリとドキュメント再生プラットフォーム(例: 実行及び閲覧)の間、リポジトリとクレジットサーバの間、又はリポジトリとオーソライゼーションサーバの間で発生する。トランザクションが一つ以上のリポジトリ同士の間で発生する時、これらのリポジトリ同士に信頼できる通信チャネルがあると仮定される。この通信チャネルは、例えば、転送エラーを検出し、補正するための組込み(built-in) 機能を有するTCP/IPチャネル又は他の市販されているチャネルであってもよい。しかしながら、通信チャネルが安全であるとは仮定されない。セキュリティ及びプライバシーの提供は、リポジトリを指定、実行するための必要条件の一部であり、これにより、種々のトランザクションに対する必要性が構成される。

【0123】トランザクションはリポジトリ同士の間いくつかの通信があることを要求する。リポジトリ同士の間通信はメッセージと称されるユニットにおいて発生する。通信ラインは安全でないと仮定されるので、最低セキュリティクラスより上にある全てのリポジトリとの通信は、パブリックキー暗号技術を用いて暗号化される。パブリックキー暗号化は暗号化技術において周知の技術である。用語「キー」は暗号化及び復号のアルゴリズムと共に使用される。キーはペアで現れ、そこでは、「書き込みキー」がデータを暗号化するために使用され、「チェックキー」がデータを復号するために使用される。書き込み及びチェック・キーは共にパブリック又はプライベートであってもよい。パブリックキーは他に配給されるキーである。プライベートキーは秘密保持される。

【0124】キーの管理とセキュリティはパブリックキー暗号化システムの成功において有用である。本発明の好ましい実施の形態において、一つ以上のマスタリポジトリはこれらのキーを保持し、リポジトリによって使用される識別証明書を作成する。

【0125】発送リポジトリが受信リポジトリにメッセージを転送する時、この発送リポジトリは、当該受信リ

ポジトリのパブリックな書き込みキーを用いてそのデータの全てを暗号化する。発送りポジトリは、それ自体の名前、受信リポジトリの名前、(以下に記述される)臨時語(nonce)のようなセッション識別子、及び各メッセージにおけるメッセージカウンタを含む。

【0126】このように、通信は、復号に対してプライベートなチェックキーを保持する受信リポジトリによってのみ(高い確率で)読み出されることができる。補助データはセキュリティを様々な再生攻撃から保護するために使用される。メッセージが誤ったカウンタ又は古い臨時語と共に到着した場合、リポジトリは、誰かが通信を妨害し、トランザクションが終了したと仮定することができる。

【0127】暗号化のために使用されるリポジトリのためのそれぞれのパブリックキーは以下に記述される登録トランザクションにおいて得られる。

【0128】使用トランザクションはリポジトリ同士の間のセッションにおいて実行される。一つ以上のリポジトリを含む使用トランザクションに対して、又はリポジトリとクリジットサーバの間の金融トランザクションに対して、登録トランザクションが実行される。ログイントランザクションと呼ばれる第2のトランザクションが、セッションを開始するために要求されることもある。リポジトリ同士の間の通信チャネルは信頼できるが、安全ではないと仮定されるので、非リポジトリがリポジトリへの合法的でないアクセスを得るためにプロトコルを模倣するかもしれない危険性がある。

【0129】二つのリポジトリ間の登録トランザクションは、図16及び17に関して記述されている。記述されているステップは、識別を「リポジトリ2」に登録する「リポジトリ1」の観点からである。その識別をリポジトリ1に登録しているリポジトリ2に対して同じセットのステップが反復されるよう、登録は対称的でなければならない。図16に関しては、ステップ1601においてリポジトリ1は最初に暗号化された登録識別子を発生し、ステップ1602において登録メッセージを発生する。登録メッセージは、マスタリポジトリの識別子、リポジトリ1に対する識別証明書、及び暗号化されたランダム登録識別子で構成されている。識別証明書は、そのプライベートキーでマスタリポジトリによって暗号化され、当該リポジトリ(ここではリポジトリ1)が本物のリポジトリであることを証明する。識別証明書はリポジトリ、リポジトリセキュリティレベル、及び(その時間を過ぎれば証明書がもはや有効ではないことを示す)タイムスタンプのためのパブリックキーも含む。登録識別子はこの登録のためのリポジトリによって発生された数字である。当該登録識別子はセッションに対して唯一であり、リポジトリ1のプライベートキーで暗号化される。当該登録識別子はいくつかの種類の通信ベース攻撃を検出することによって認証のセキ

ュリティを改善するために使用される。次に、ステップ1603において、リポジトリ1は、登録メッセージをリポジトリ2へ転送する。

【0130】登録メッセージを受け取ると、ステップ1604において、リポジトリ2は当該メッセージがマスタリポジトリに必要とされるパブリックキーを持っているか否かを決定する。リポジトリ2が識別証明書を復号するのに必要とされるパブリックキーを持たない場合、ステップ1618において、登録トランザクションはエラーで終了する。

【0131】リポジトリ2が適当なパブリックキーを有すると仮定すると、ステップ1605において、識別証明書が復号される。ステップ1606において、リポジトリ2は暗号化された登録識別子を保存し、ステップ1607において、リポジトリ識別子を抽出する。ステップ1608において、抽出されたリポジトリ識別子は、採用されたドキュメントリポジトリの「ホットリスト」に対してチェックされる。本発明の好ましい実施の形態において、各リポジトリは採用されたリポジトリの「ホットリスト」を含む。そのリポジトリが「ホットリスト」に載っている場合、ステップ1618において、登録トランザクションはエラーで終了する。リポジトリはそれらの証明書の期限が切れると、ホットリストから除外されることができ、これによりリストは際限なく成長しなくてもよい。さらに、リポジトリは以前に受け取ったホットリスト証明書のショートリストを保持することによって、当該リポジトリは実際にこのリストを検討する作業を避けることができる。これらのリストはマスタリポジトリによって暗号化される。能率を改善するためのアプローチのためのマイナーな変形は、ホットリスト証明書の名前のリストを最初に交換するリポジトリを有し、結果的に、以前に受け取ったことのないリストのみが交換される。「ホットリスト」はマスタリポジトリによって保持され、配給される。

【0132】トランザクションがエラーで終了する代わりに、トランザクションが、他の登録メッセージが他のマスタリポジトリによって作成された識別証明書に基づいて送られることを要求することができることに注目されたい。これは、申し分のない識別証明書が見つかるまで又は信頼が確立されないことが決定するまで反復される。

【0133】リポジトリがホットリスト上にないと仮定した場合、リポジトリ識別は検証されなければならない。言い換えれば、リポジトリ2は、(リンクの)他端におけるリポジトリが実際にリポジトリ1であることを検証する必要がある。これはパフォーマンステストと呼ばれ、リポジトリ1とリポジトリ2の間の前のセッション開始のレコードを再生する偽リポジトリを介してリポジトリへの無効なアクセスを防止するために実行される。パフォーマンステストは、ステップ1609

においてパフォーマンスメッセージを発生するリポジトリ-2によって開始される。パフォーマンスメッセージは、臨時語（ノンス）、それぞれのリポジトリの名前、時間、及びリポジトリ-1から受け取られた登録識別子からなる。ノンスは、いくつかの任意の可変情報（例えば、時間又は温度）に基づいて生成されたメッセージである。このノンスは、リポジトリ-1がメッセージ上で今まで一度も見ることが要求されるプライベートキーを用いて、当該リポジトリ-1がメッセージの正確な暗号化を実際に展開することができるかをチェックするために使用される。パフォーマンスメッセージはリポジトリ-1の登録メッセージ内で指定されたパブリックキーを用いて暗号化される。このパフォーマンスメッセージは、ステップ1610において、リポジトリ-1へ転送され、ステップ1611において、そのプライベートキーを用いてリポジトリ-1によって復号される。次に、リポジトリ-1は、ステップ1612において二つのリポジトリの名前が正しいか、ステップ1613において時間が正確か、ステップ1614において登録識別子がリポジトリ-1が送った識別子に対応しているか否かを確認するためにチェックする。これらのテストのうちのどれかが不合格ならば、トランザクションはステップ1616で終了する。これらのテストが合格すれば、ステップ1615において、リポジトリ-1はノンスをリポジトリ-2へ暗号化しないで転送する。次に、リポジトリ-2は、ステップ1617において、受け取ったノンスを元のノンスと比較する。これらが一致しない場合、ステップ1618において、登録トランザクションはエラーで終了する。これらが同じ場合は、登録トランザクションは無事に終了する。

【0134】この時、このトランザクションが終了しなかったと仮定すると、これらのリポジトリは、セッションの間の全ての通信において使用されるセッションキーを含むメッセージを交換し、クロックを同期する。図17は、セッション情報交換及びクロック同期ステップを示す（再度、リポジトリ-1の観点から）。図17に関しては、ステップ1701においてリポジトリ-1はセッションキーのペアを作成する。第1のキーは、プライベートに保持され、メッセージを暗号化するためにリポジトリ-1によって使用される。第2のキーはリポジトリ-2によって使用されるパブリックキーであり、メッセージの復号を行う。この第2のキーは、ステップ1702において、リポジトリ-2のパブリックキーを用いて暗号化され、ステップ1703において、リポジトリ-2へ送られる。受け取ったら、ステップ1704において、リポジトリ-2は当該第2のキーを復号する。第2のキーは次に続く通信においてメッセージを復号するために使用される。各リポジトリがこのステップを終了すると、これらのリポジトリは共に、もう一方のリポジトリが本物であり、これらのリポジトリがオリジナルの

リポジトリと通信していることが確認される。各リポジトリはセッション中の更なる通信を復号するときに使用されるキーをもう一方のリポジトリへ付与する。このキーはそれ自身が受信リポジトリのみのパブリックキーで転送されるので、引き続きメッセージを復号するために使用されるキーを復号することが可能である。

【0135】セッションの情報が交換された後、リポジトリはこれらのクロックの同期をとらなければならない。クロックの同期は、これらのリポジトリの互いのトランザクションの金融記録に対して合意が得られたタイムベースを設定するために当該リポジトリによって使用される。図17に戻って、ステップ1705において、リポジトリ-2は、タイムスタンプ交換メッセージを発生することによって、クロック同期を開始し、ステップ1706においてこのメッセージをリポジトリ-1へ転送する。受信すると、ステップ1707においてリポジトリ-1はそれ自体のタイムスタンプメッセージを発生し、ステップ1708においてそのメッセージをリポジトリ-2へ戻すように転送する。ステップ1709において、リポジトリ-2は現在時間を記し、ステップ1710において、リポジトリ-1から受け取った時間を記憶する。ステップ1711において、現在時間がリポジトリ-1から受け取った時間と比較される。ステップ1712において、この差は、所定の許容値（例えば、1分）を超過しているかを見るためにチェックされる。超過している場合、ステップ1713において、リポジトリ-2は、当該リポジトリ-2がリポジトリの偽造を示すかもしれないのでトランザクションを終了する。超過していない場合、ステップ1714において、リポジトリ-2は調整されたタイムデルタを計算する。調整されたタイムデルタは、リポジトリ-2のクロックタイムとリポジトリ-1及びリポジトリ-2からの時間の平均との差である。

【0136】より高い精度を達成するために、リポジトリ-2は時間を一定回数（例えば、5回）まで再び要求し、クロック同期ステップを反復し、その結果を平均する。

【0137】第2のセッション開始トランザクションはLogin transaction（ログイン・トランザクション）である。Login transactionはトランザクションを要求するユーザの真偽をチェックするために使用される。Login transactionは、クレジットサーバへ請求される金融トランザクションのオーソライゼーションに関して特に慎重である。Login transactionは、ユーザインタフェースにおけるユーザとリポジトリに対応しているクレジットサーバとの対話を含む。ここで交換される情報は、ユーザに対して自体を識別するためにリポジトリ/クレジットサーバによって供給されるログインストリングと、クレジットサーバに対して自身を識別するためにユーザによって提供されるPersonal Identification Numb

er(PIN) (個人識別番号)である。ユーザが、ユーザインタフェースが常駐しているリポジトリとは異なるリポジトリでクレジットサーバへアクセスしている場合、情報の交換はそれぞれのリポジトリのパブリック及びプライベートキーを用いて暗号化される。

【0138】Billing Transaction (請求トランザクション)はクレジットサーバとの金銭上のトランザクションに関する。請求トランザクションは全ての他の条件が満たされた時に実行され、この要求を許諾するために使用料が要求される。大抵の場合、請求トランザクションは当該技術分野においては良く理解される。これらのトランザクションは、リポジトリとクレジットサーバの間、又はクレジットサーバと請求クリアリングハウスの間にある。つまり、要求されるトランザクションは以下を含む：

- ・リポジトリとユーザがそれらの本物をクレジットサーバへ設定するための登録及びLOGIN トランザクション。リポジトリとクレジットサーバが単一システムとして実行される場合、これらのトランザクションは完全に内部トランザクションである。

- ・クレジットサーバがその本物を請求クリアリングハウスへ設定する登録及びLOGIN トランザクション。

- ・料金を割り当てるためのAssign-fee transaction。このトランザクションにおける情報は、トランザクション識別子、トランザクションにおけるリポジトリの同一性、及びデジタルワークの該当部分からの請求のリストを含む。トランザクションにおいて何らかの異常事態、例えば、通信妨害などが発生した場合、この情報も含まれる。

- ・請求を割り当てるためのBegin-charge transaction (請求開始トランザクション)。このトランザクションは、課金使用に用いられることを除いては料金割当てトランザクション (Assign-fee transaction) とほぼ同様である。なお、このトランザクションは、料金割当てトランザクションと共に使用料情報を含む。次に、クレジットサーバはクロックをランする役割を果たす。

- ・課金使用に対する請求を終了するEnd-charge transaction。(このアプローチの変形ではリポジトリは時間ブロック毎に周期的請求情報を交換する。)

- ・パーソナルクレジットサーバと請求クリアリングハウスの間のReport charge transaction (請求通知トランザクション)。このトランザクションは請求(billing)期間において少なくとも一度は呼び出される。これは、請求に関する情報を渡すために使用される。貸借カード上で、このトランザクションは、必要に応じて、バランス(貸借対照)情報及びクレジットの制限を更新するために使用される。

【0139】全ての請求トランザクションにはトランザクションIDが付与され、サーバとクライアントの両方によってクレジットサーバへ通知される。これによ

て、トランザクションのパーティの一人が銀行カードを無くした場合、請求情報の損失の可能性を軽減し、システムの偽造に対するチェックを行う。

【0140】セッション開始トランザクションが終了した後で、使用の要求が処理されてもよい。使用要求を処理する時に実行されるステップの記述を簡素化するため、用語「リクエスト」は要求を開始するリクエストモードのリポジトリを称するために使用される。用語「サーバ」は、サーバモードのリポジトリを称するために使用されると共に、所望されるデジタルワークも含む。ワークをプリントするか又は閲覧するかの要求のような多数のケースにおいては、リクエスト及びサーバは同じデバイスであってもよいし、以下に記述されているトランザクションは完全に内部トランザクションであってもよい。このようなインスタンスにおいて、登録トランザクションのようないくつかのトランザクションステップが実行されなくてもよい。

【0141】使用権トランザクションの全てのセマンティックス(意味論)の一部であるいくつかの共通のステップがある。これらのステップは共通トランザクションステップと呼ばれる。「オープニング」ステップと「クロージング」ステップの二つのステップがある。簡素化のために、これらは使用権トランザクションの全ての記述においてそれらを反復するよりも、本発明においてはリストされている。

【0142】トランザクションは、デジタルワークの一部、完全なデジタルワーク、又は他のデジタルワークを含むDigital workを指示することができる。本明細書中には詳細に記述されていないが、トランザクションは複数のデジタルワークからなるフォルダさえも指示することができる。用語「ワーク」はアクセスされているデジタルワークの任意の部分又は任意のセットを指示するために使用される。

【0143】ステップの多くは特定の条件が満たされているか否かを決定することを含む。各使用権は、権利が行使される前に満たされなければならない一つ以上の条件を有することもあることを思い出して頂きたい。デジタルワークは部分を有し、部分は部分を有する。異なる部分は異なる権利及び料金を有することができる。従って、トランザクションに含まれる全ての部分に対して必要条件が満たされることを検証することが必要である。要するに、権利が存在しているか及び行使するための条件が満たされているかのチェックに指示される時、全てのこのようなチェックがワークの関連部分のそれぞれに対して発生するように意図されている。

【0144】図18はトランザクションのための最初の共通のオープニング及びクロージングステップを示している。この時点で、登録が発生し、「信頼できる(trusted)」セッションが適所に置かれていると仮定される。

汎用テストは、ワークを含むフォルダ又はファイルシ

テム階層内の高位にあるフォルダを含む幾つかのフォルダに対応している使用権に対するテストである。これらのテストは、ワークがワークそれ自体にアタッチされていることに対抗して、ワークが特定のリポジトリ上にあった結果としてのワークに課せられた必要条件に対応している。図18に関しては、ステップ1801において、使用トランザクションを開始する前に、リクエストは当該トランザクションに対応している権利が行使される前に、要求されるあらゆる汎用テストを実行する。例えば、(ソフトウェアの)インストール権、アンインストール権及びデリート権が、権利が行使される前にリクエストがオーソライゼーション証明書を有することを要求するために実行されてもよい。他の例は、デジタルワークがリクエストへコピーされる前に、デジタルチケットが提示され、パンチされるという必要条件である。汎用テストのいずれかが不合格ならば、ステップ1802において、トランザクションは開始されない。使用要求を受けて、このように要求されたテストに合格したと仮定すると、ステップ1803において、サーバは、トランザクションの記録又は通知に使用されるトランザクション識別子を発生する。次に、サーバは、ステップ1804において、要求されたトランザクションに対応している権利がデジタルワークに対して許諾されたか否かをチェックする。当該要求に対応している権利がデジタルワークに対して許諾されなかった場合、ステップ1805においてトランザクションは終了する。要求された権利がデジタルワークに対して許諾された場合、サーバは、権利を行使するための種々の条件が満たされたか否かを決定する。ステップ1806において、タイムベースの条件が検討される。これらの条件は権利のバージョンに対するタイム仕様を検討することによってチェックされる。いずれの条件も満たされなかった場合、ステップ1805においてトランザクションは終了する。

【0145】タイムベースの条件が満たされた場合、ステップ1807においてサーバはセキュリティ及びアクセス条件をチェックする。このようなセキュリティ及びアクセス条件は以下の場合に満たされる。

- 1) リクエストが指定されたセキュリティクラス又はそれより高位のセキュリティクラスにあること。
- 2) サーバがあらゆる指定されたオーソライゼーションテストに合格していること。
- 3) リクエストがあらゆる指定されたオーソライゼーションテストに合格し、あらゆる要求されたデジタルチケットを有していること。

いずれの条件も満たされなかった場合、ステップ1805においてトランザクションは終了する。

【0146】セキュリティ及びアクセス条件がすべて満たされたと仮定すると、ステップ1808においてサー

バはコピーカウント条件をチェックする。コピーカウントがゼロに等しい場合、トランザクションは完了することができず、ステップ1805においてトランザクションは終了する。

【0147】コピーカウントがゼロに等しくないと仮定すると、ステップ1809において、サーバは、要求された権利のために使用されているコピーが、要求された権利(又は関連部分)のためのコピーカウントに等しいか又はそれ以上であるか否かをチェックする。使用中のコピーがコピーカウントより大きい場合又は等しい場合、これはトランザクションのバージョンに対する使用権が使い果たされたことを示す。従って、ステップ1805において、サーバはトランザクションを終了する。コピーカウントがトランザクションに使用されているコピーより少ない場合、トランザクションは続けられ、ステップ1810においては、使用されているコピーが当該トランザクションにおいて要求されたデジタルワークの数だけインクリメントされる。

- 【0148】次に、サーバは、ステップ1811においてデジタルワークが“Loan”アクセス権利を有しているか否かをチェックする。“Loan”アクセス権利は、全てのコピーが貸し出されても権利が残存していることがあるので、特別のケースである。デジタルワークが“Loan”アクセス権利を有している場合、ステップ1812において、全てのコピーが貸し出されたか否かを見るためにチェックされる。貸し出されたコピーの数はデジタルワークの貸出権利の全てのバージョンに対するCopy-Countsの合計である。複合ワークに関しては、関連する数字は、複合ワークの各構成要素の合計のような最小数である。全てのコピーが貸し出された場合、ステップ1813において、残りの権利が決定される。残りの権利は、“Loan”の権利のバージョンからの残りの権利仕様から決定される。“Loan”の権利が一つだけの場合、決定は簡単である。残りの権利はLoanの権利のバージョンにおいて仕様された権利か、又はRemaining-Rights: が指定されていない場合は無しである。Loanの権利の複数バージョンがある場合、及び全てのバージョンの全てのコピーが貸し出された場合、残りの権利は、Loanの権利の全てのバージョンに渡って残っている権利の最小セット(共通部分)として取られる。ステップ1814において、サーバは要求されたセットが残りの権利のセット内にあるか否かを決定する。要求されたセットが残りの権利のセット内にない場合、ステップ1805において、サーバはトランザクションを終了する。

【0149】Loanがデジタルワークの使用権ではない場合、又は全てのコピーが貸し出されず、又は要求された権利が残りの権利のセット内にある場合、ステップ1815において、その権利に対する料金の条件がチェックされる。これはリポジトリとそれに対応するクレジットサーバの間の種々の金融トランザクションを開始す

る。さらに、デジタルワークの使用のあらゆる課金が開始される。任意の金融トランザクションが失敗すると、ステップ1805においてトランザクションは終了する。

【0150】条件がチェックされる順番がステップ1806乃至1815の順序に従わなくともよいことに注目されたい。

【0151】この時点で、権利特定ステップは今や実行され、ステップ1816として表される。この権利特定ステップは以下により詳細に記述されている。

【0152】共通のクロージングトランザクションステップが実行される。クロージングトランザクションステップの各々は、トランザクションが無事に完了した後で、サーバによって実行される。図18に戻ってみると、ステップ1817において、要求された権利に対して使用されているコピーの値は当該トランザクションに含まれているコピー数だけデクリメントされる。次に、権利が計量による使用料金仕様を有する場合、ステップ1818において、サーバはトランザクションに含まれている全ての部分に対する権利に対応しているRemaining-Use-Timeから経過した時間を減算する。最後に、権利に対応している料金仕様がある場合、ステップ1819において、サーバは請求を確認するためにEnd-Charge（請求終了）金融トランザクションを開始する。

【0153】考慮すべき重要な部分は、サーバからリクエストへのデジタルワークの転送である。本明細書中に記述されている転送プロトコルは有効セッションが作成された後で発生する事象に関する。転送プロトコルは、リポジトリ同士の間での通信中断ケースを取り扱わなければならない。通信チャネルへのノイズ注入のような妨害は完全性チェック（例えば、パリティ、チェックサム等）によって検出されることができる。当該完全性チェックは、伝送プロトコルの中へ組み込まれているが本明細書中では詳細に説明しない。

【0154】転送プロトコルの基盤となる目標は、通信チャネルの故意又は事故による妨害のようないくつかの故障モードを阻止することである。例えば、トランザクションの終りに近い特定の時間においてユーザがクレジットサーバを有するカードを引いたと仮定する。「カードを引く」ことによってリポジトリが作成されたワークのコピー数を正確に数え損なうような無防備な時間があってはならない。要するに、デジタルワークを使った後で、支払いを避けるための手段としてパーティが接続を破壊できるような時間があってはならない。

【0155】トランザクションが妨害（及び失敗）された場合、両方のリポジトリは、デジタルワークと、失敗前の当該デジタルワークの状態に対する会計、及び失敗それ自体のモジュール記録を回復する。

【0156】図19は、トランザクション中に情報を転送する過程におけるステップを示す状態ブロックであ

る。各ボックスは、サーバモード（中央の点線1901の上）又はリクエストモード（点線1901の下）のいずれかにおいてリポジトリの状態を示す。実線の矢印は状態同士間の遷移を表す。ダッシュラインの矢印は、リポジトリ同士間のメッセージ通信を表す。実線の遷移矢印を指し示すダッシュラインのメッセージ矢印は、メッセージが受け取られた時、遷移が発生することを意味すると解釈される。ラベル付けされていない遷移の矢印は、無条件で発生する。状態遷移矢印における他のラベルは遷移をトリガーする条件を記述する。

【0157】図19に関しては、サーバは最初に新しいトランザクションがスタートメッセージ1903を介して開始される1902の状態にある。このメッセージはトランザクション識別子と、転送されるデータのブロックのカウントとを含むトランザクション情報を含む。最初は、1904において待機状態にあるリクエストはデータ待機状態1905を実行する。

【0158】サーバはデータ転送状態1906を実行し、データブロック1907を転送し、次にアクノレージ状態1908の待機を実行する。データが受け取られると、リクエストはデータ受け取り状態1909を実行し、データブロックが完全に受け取られた時それはアクノレージ状態1910を実行し、Acknowledgement message（アクノレージ・メッセージ）1911をサーバへ転送する。

【0159】より多くのブロックを送る場合、サーバはリクエストからAcknowledgement messageを受け取るまで待機する。Acknowledgement messageが受け取られると、サーバは次のブロックをリクエストへ送り、アクノレージを再び待機する。このリクエストも同じ状態サイクルを反復する。

【0160】サーバが最終ブロックを送る前に通信の失敗を検出した場合、当該サーバはトランザクションが取り消されるキャンセル状態1912に入る。同様に、リクエストが最終ブロックを受け取る前に通信の失敗を検出した場合、当該リクエストは取消状態1913に入る。

【0161】送ろうとするブロックがなくなった場合、サーバはトランザクションにコミット（委任）し、状態1914における最終のAcknowledgement（アクノレージ）を待機する。サーバが最終のAcknowledgement messageを受け取る前に通信の失敗があった場合、サーバはやはりトランザクションにコミットするが、状態1915におけるそのクレジットサーバへの事象に関する通知を含む。この通知は二つの目的を果たす。この通知は、完全には受け取られなかったデジタルワークを受け取ったものとして請求されたことについてのユーザによる主張を合法とするのを補助する。さらに、これは、疑わしい使用パターン及び妨害を有するリポジトリや通信ラインを識別するのを補助する。サーバは、次に、その完

了状態1916を実行する。

【0162】受け取ろうとするブロックがもうないリクエストサイドでは、リクエストは状態1917においてこのトランザクションにコミットする。リクエストがこの状態における通信の失敗を検出すると、リクエストは状態1918におけるその失敗をクレジットサーバへ通知するが、依然としてそのトランザクションにコミットし続ける。リクエストがコミットした時、それはアクノレッジメッセージをサーバへ送る。サーバは、次に、その完了状態1919に入る。

【0163】キー特性は、サーバ及びリクエストの両方が、全てのデータブロックが引き渡される前にトランザクションが妨害された場合にトランザクションをキャンセルすると共に、全てのデータブロックが引き渡された場合に当該トランザクションにコミットすることである。

【0164】サーバが全てのデータブロックを送った（及びコミットした）が、リクエストは全部を受け取っていないかったり、そのトランザクションをキャンセルするような可能性がある。この場合、恐らくは、両方のリポジトリが通信の失敗を検出し、それをこのリポジトリのクレジットサーバへ通知する。通信の失敗の非常に正確なタイミングに依存するのでこのケースは恐らくは稀である。唯一の結果は、リクエストリポジトリにおけるユーザは、クレジットサービスからの払戻しを望み、この払戻しに対するケースは両リポジトリによる通知により実証されることである。

【0165】データの損失を防止するために、サーバは、リクエストからの最終アクノレッジを受け取るまで、あらゆる転送されたデジタルワークをデリートしてはならない。しかし、サーバ又はファイルを使用してもならない。この状態を処理する周知の方法は、「2相コミット」又は2PC（2 phase commit）と呼ばれる。

【0166】「2相コミット」は以下のように作用する。第1フェーズは上記に示された方法と同様に作用する。サーバはリクエストへ全てのデータを送る。両方のリポジトリは、トランザクション（及び適切なファイル）にコミットされなかったものとしてマークする。サーバは、ready-to-commit メッセージをリクエストへ送る。リクエストはアクノレッジをリターンする。次に、サーバはコミットし、リクエストにコミットメッセージを送る。リクエストがコミットメッセージを受け取った時、リクエストはファイルにコミットする。

【0167】通信の失敗又は他のクラッシュがあった場合、リクエストはトランザクションの状況を決定するためにサーバと共にチェックバックしなければならない。サーバはこれに対して最終ワードを有している。リクエストが全てのデータを受け取っているかもしれないが、リクエストが最終メッセージを得ていなかった場合、リクエストはコミットしてはいない。サーバが一旦コミッ

トすると、ファイルが完全に転送されたことが2PCサイクルを開始する前にわかっているため、サーバは先へ進んで（トランザクション記録を除いて）ファイルをデリートすることができる。

【0168】同じ効果を達成するために使用されることができる技術的に公知の変形がある。例えば、サーバがワークをクライアントへ転送する時に付加的暗号化レベルを使用することができる。クライアントがメッセージアクノレッジの受領情報を送った後でのみ、このクライアントはキーを送る。次に、クライアントはデジタルワークに対しての支払いに合意する。この変形のポイントは、クライアントがワークを受け取ったことの明白な会計履歴を提供することにある。しかしながら、信用のあるシステムに関しては、この変形は暗号化レベルの追加によって会計処理能力の実利得をもたらさない。

【0169】特定の使用権に対するトランザクションを以下に説明する。Copy Transaction（コピートランザクション）は、同等か又はやや少ない使用権を有するワークの一つ以上の独立コピーを作成するための要求である。コピーは、それがデジタルワーク全体又はデジタルワークを含むフォルダ全体を指示する点で後で説明される抽出権とは異なる。コピー動作はデジタルワークのある部分を取り除くために使用することはできない。

・リクエストはサーバへCopy Transactionを開始するようにメッセージを送る。このメッセージはコピーされるワーク、トランザクションに使用される著作権のバージョン、ワークを配置するための宛て先アドレス（フォルダ内の位置）情報、ワークに対するファイルデータ（サイズを含む）、要求されたコピー部数を示す。

・リポジトリは共通のオープニング・トランザクション・ステップを実行する。

・サーバは、転送プロトコルに従って、要求された内容及びデータをクライアントへ転送する。Next-Set-Of-Rightsが権利のバージョンに提供された場合、これらの権利がワークに対する権利として転送される。そうでない場合、オリジナルの権利が転送される。あらゆる場合において、送られているデジタルワークのコピーに対するCopy-Countフィールドが、要求されたコピー部数にセットされる。

・リクエストは、ワークの内容、データ、及び使用権を記録し、このワークを記憶する。リクエストはコピーが作成された日付と時間をデジタルワークの属性の中に記録する。

・リポジトリは、共通のクロー징トランザクションステップを実行する。

【0170】Transfer transaction（転送トランザクション）は、同等又はやや少ない使用権を有するワークのコピーを他のリポジトリへ移動するための要求である。

50 Copy Transactionとは対照的に、これはサーバからワー

クコピーを除去することになる。

・リクエストはサーバへTransfer transactionを開始するようにメッセージを送る。このメッセージは転送されるワーク、当該トランザクションにおいて使用される転送権のバージョン、ワークを配置するための宛て先アドレス情報、このワークのためのファイルデータ、及び含まれるコピー数を示す。

・リポジトリは共通のオープニングトランザクションステップを実行する。

・サーバは、転送プロトコルに従って、要求された内容及びデータをリクエストへ転送する。Next-Set-Of-Rightsが権利のバージョンに提供された場合、これらの権利がワークに対する権利として転送される。そうでない場合、オリジナルの権利が転送される。両方のケースにおいて、転送された権利に対するCopy-Countフィールドは、要求されたコピー部数にセットされる。

・リクエストはワークの内容、データ、及び使用権を記録し、このワークを記憶する。

・サーバは当該トランザクションに含まれるコピー数だけそのコピーカウントをデクリメントする。

・リポジトリは共通のクロー징トランザクションステップを実行する。

・サーバ内に残っているコピー数が現在ゼロである場合、デジタルワークはメモリから消去される。

【0171】Loan transaction（貸出しトランザクション）はデジタルワークのコピーを貸し出すためのメカニズムである。この貸出しの最大期間はデジタルワークの内部パラメータによって決定される。ワークは所定期間の後で自動的に返却される。

・リクエストはサーバへLoan transactionを開始するようにメッセージを送る。

このメッセージは貸し出されるワーク、当該トランザクションにおいて使用される貸出権のバージョン、ワークを配置するための宛て先アドレス情報、包含されるコピー数、ワークのためのファイルデータ、及び貸出しの期間を示す。

・サーバは、要求された貸出期間の有効性をチェックし、この期間が有効でない場合は、エラーで終了する。貸出されたコピーに対する貸出しは、サーバへのオリジナルの貸出期間を超過することができない。

・リポジトリは共通のオープニングトランザクションステップを実行する。

・サーバは、要求された内容及びデータをリクエストへ転送する。Next-Set-Of-Rightsが提供された場合、これらの権利がワークに対する権利として転送される。そうでない場合、オリジナルの権利が、貸出期間を反映するように更新されてから、転送される。

・リクエストはワークの内容、データ、使用権、及び貸出期間を記録し、このワークを記憶する。

・サーバは貸し出されたコピー数を反映するようにディ

ジタルワークにおける使用権情報を更新する。

・リポジトリは、共通のクロー징トランザクションステップを実行する。

・サーバは、デジタルワークに対する使用権データを更新する。これは、ワークが返却されるまで、このワークの使用を除外する。リクエストプラットフォームにいるユーザは、ここで、デジタルワークの転送されたコピーを使用することができる。オリジナルリポジトリにアクセスしているユーザは、コピーが残っていない限り、デジタルワークを使用することができない。次に

発生することは、時間における事象の順序に依存する。

ケース1：貸出し期間の時間はまだ使い切られておらず、リクエストがリポジトリへReturn message（返却メッセージ）を送る場合、

・返却メッセージはリクエスト識別、及びトランザクションIDを含む。

・サーバは、返却されたコピー部数だけ使用中のコピーをデクリメントする。（返却されたデジタルワークの数が実際に貸し出された数より大きい場合はエラーとして処理される。）このステップは、ここで、他のユーザに対するサーバにおいてワークを有効にすることができ

る。
・リクエストはそのコピーを無効にして、メモリからその内容を除去する。リクエストは全ての現在使用を終了し、メモリからデジタルワークコピーを消去する。いずれにしろワークは自動的に返却されるのであるが、リクエストは貸出し期間より早くワークを返却することが想定される。早めの返却理由の一つは、貸出しコストを決定する計量による料金の存在である。早期返却はこの料金を削減することができる。

ケース2：貸出し期間の時間が使い果たされ、リクエストがまだReturn messageを送っていない場合、

・サーバは、使用中のコピーフィールドを、貸し出されたデジタルワークの数だけデクリメントする。

・リクエストは、デジタルワークのコピーを自動的に無効にする。当該リクエストは全ての現在使用を終了し、メモリからデジタルワークコピーを消去する。ワークはいずれにしろ自動的に返却されるのにもかかわらず、リクエストは貸出し期間より早くワークを返却することが想定される。早期返却理由の一つは、貸出しコストを決定する計量による料金の存在である。早期返却はこの料金を削減することができる。

【0172】Play Transaction（再生トランザクション）はワークの内容を使用するための要求である。一般に、ワークを「再生（play）」することは、スピーカ又はディスプレイデバイスのようなある種類のトランスデューサを介してデジタルワークへ送ることである。この要求は、これらの内容が他のシステムへデジタル式に通信されないように意図されていることを示唆している。例えば、これらの内容はプリンタへ送られるか、デ

ィジタル媒体に記録されるか、トランザクション後に保持されるか、又は他のリポジトリへ送られる。

【0173】用語「再生(Play)」は、音楽の再生、映画の再生、又はビデオゲームの再生などの例に対して自然である。再生の一般的な形態は、ディジタルワークを使用するために「プレーヤー」が使用されることを意味する。しかしながら、この用語「再生」は、全ての記録媒体及び記録種類を網羅する。従って、ある人がディジタルワークを「再生」するとは、それを読み出すためにレンダリングすることを意味し、又はコンピュータプログラムを「再生」するとは、それを実行することを意味する。ディジタルチケットに関しては、プレーヤーはディジタルチケットエージェントにもなり得る。

・リクエストはサーバへPlay Transactionを開始するようにメッセージを送る。このメッセージは、再生されるワーク、当該トランザクションにおいて使用される再生権のバージョン、使用されているプレーヤーの識別、及びワークに対するファイルデータを示す。

・サーバは、プレーヤーの識別の有効性と、このプレーヤー識別と権利上のプレーヤー仕様の互換性をチェックする。これらが満たされない場合、エラーで終了する。

・リクエストは、共通のオープニングトランザクションステップを実行する。

・サーバ及びリクエストは、転送プロトコルに従って、プレーヤーによって要求された通りに、データのブロックの読み出し、書き込みを行う。リクエストはプレーヤーを用いてワークの内容を再生する。

・プレーヤーが終了した時、プレーヤーとリクエストはそれらのメモリから内容を除去する。

・リポジトリは共通のクロー징トランザクションステップを実行する。

【0174】Print transaction (プリント・トランザクション) はワークの内容を「プリンタ」上でレンダリング(描画)するために当該ワークの内容を得るための要求である。我々は、インクで用紙に書き込む一般的なケースを含むように、用語「プリンタ」を使用する。しかしながら、我々のこの用語の使用における「プリンティング」の主な態様は、使用権の保護の外側にある場所で、ディジタルワークのコピーを作成することにある。全ての権利と同様に、これは特別のオーソライゼーション証明書を要求してもよい。

【0175】ディジタルワークが一旦プリントされると、発行者及びユーザは、どんな著作権法が効力があるかによって制限される。しかしながら、プリンティングはそれらの内容をリポジトリの制御外へ移動する。例えば、他の実施メカニズムがない場合、ディジタルワークが一旦用紙にプリントされると、ディジタルワークは使用料金を集金するリポジトリによって妨害されずに通常の複写機でコピーされることができ得る。ディジタルディスクへのプリンタが許容された場合、このディジタルコ

ピーは使用権の制御外にある。クリエイタはこのような著作権法を侵すコピーに絶対に暗黙に同意しないが、クリエイタとユーザは共にこのことを知っている。

・リクエストは、サーバへPrint transactionを開始するようにメッセージを送る。このメッセージは再生されるワーク、使用されるプリンタの識別、ワークのためのファイルデータ、及び要求されているコピー数を示す。

・サーバは、プリンタ識別の有効性、及びプリンタ識別と権利上のプリンタ仕様の互換性をチェックする。これらの条件が満たされない場合、エラーで終了する。

・リポジトリは共通のオープニングトランザクションステップを実行する。

・サーバは、転送プロトコルに従って、データのブロックを転送する。

・リクエストは、プリンタを用いて、ワーク内容をプリントする。

・プリンタが終了した時、プリンタとリクエストはそれらのメモリからこれらの内容を除去する。

・リポジトリは共通のクロー징トランザクションステップを実行する。

【0176】Backup transaction (バックアップ・トランザクション) は、媒体の故障に対する保護としてディジタルワークのバックアップコピーを作成するための要求である。リポジトリの内容において、セキュリティバックアップコピーは、以下の三つの点において他のコピーとは異なる。(1) これらのセキュリティバックアップコピーはCopy transactionではなく、Backup transactionの制御下で作成される。(2) これらは正規のコピーとしてカウントしない。(3) これらは正規のコピーとして使用可能ではない。概して、バックアップコピーは暗号化されている。

【0177】バックアップコピーはこれらに割り当てられる権利に依って転送されたり又はコピーされたりするが、それらの再生、プリント又は埋め込み(embedding)のためにバックアップコピーを有用にするための唯一の方法はそれらを復元することである。

【0178】Backup operation (バックアップ動作) の出力は、ワークの内容及び記述を含む暗号化されたデータファイルと、暗号化された内容を復元する暗号化キーを有する復元ファイルとの両方である。多くの場合において、暗号化されたデータファイルは、セキュリティのための暗号化のみに頼って、この出力を保護システムの外側のディスクに「プリント」するための権利を有する。このようなファイルは、物理的に安全であり便利な所に記憶されることができ得る。復元ファイルはリポジトリ内に保持される。このファイルは、バックアップコピーの復元のために必要である。このファイルは、リポジトリ同士の間での転送のための権利を有することができ得る。

・リクエストは、サーバへBackup transactionを開始す

るようにメッセージを送る。このメッセージはバックアップされるワーク、当該トランザクションにおいて使用されるバックアップ権のバージョン、バックアップコピーを配置するための宛て先アドレス情報、及びワークに対するファイルデータを示す。

・リポジトリは共通のオープニングトランザクションステップを実行する。

・サーバは、要求された内容及びデータを、リクエストへ転送する。Next-Set-Of-Rightsが提供された場合、これらの権利がワークに対する権利として転送される。そうでない場合、オリジナルのバックアップファイルに対するデフォルトの権利セットがサーバによってセットされる。

・リクエストは、ワークの内容、データ、及び使用権を記録する。それは、次に、一回使用のキーを作成し、内容ファイルを暗号化する。それはキー情報を復元ファイル内に保存する。

・リポジトリは共通のクロージングトランザクションステップを実行する。

【0179】いくつかのケースにおいて、磁気光学的記憶装置システム又は磁気テープのようなオフライン記憶装置を守るために大量の暗号化された内容ファイルをアーカイブ（保管）することができることは便利である。非リポジトリアーカイブファイルの作成は暗号化プロセスと同じ程安全である。このような非リポジトリアーカイブ記憶装置は、「プリンティング」の形式と見なされ、特定の「アーカイブ・プリンタ」を有するプリント権によって制御される。アーカイブプリンタデバイスは、検索されることができるよう、暗号化された内容ファイル（しかし、記述ファイルではない）をオフライン保存するようにプログラムされる。

【0180】Restore transaction（復元トランザクション）は、デジタルワークの暗号化されたバックアップコピーを使用可能なコピーへ変換するための要求である。復元動作は、災害による媒体の故障を補償するために使用されるように意図されている。全ての使用権と同様に、復元権はオーソライゼーションチェックを含む料金及びアクセステストを含むことができる。

・リクエストは、Restore transactionを開始するようにサーバへメッセージを送る。このメッセージは復元されるワーク、当該トランザクションにおいて使用される復元権のバージョン、ワークを配置するための宛て先アドレス情報、及びワークに対するファイルデータを示す。

・サーバは、内容ファイルが使用可能であることを検証する（即ち、この要求に対応したデジタルワークがバックアップされている）。内容ファイルが使用可能でない場合、トランザクションはエラーで終了される。

・リポジトリは、共通のオープニングトランザクションステップを実行する。

・サーバは、復元ファイルからキーを検索する。当該サーバは、ワーク内容、データ、及び使用権の暗号を解読する。

・サーバは、転送プロトコルに従って、要求された内容及びデータをリクエストへ転送する。Next-Set-Of-Rightsが提供された場合、これらの権利がワークに対する権利として転送される。そうでない場合、オリジナルのバックアップファイルに対するデフォルトの権利セットがサーバによって転送される。

10 ・リクエストは、デジタルワークを記憶する。

・リポジトリは共通のクロージング・トランザクションステップを実行する。

【0181】Delete transaction（デリート・トランザクション）は、デジタルワーク又はリポジトリからのデジタルワークの多数のコピーをデリートする。実際的には、全てのデジタルワークがデリート権を有している。

・リクエストは、Delete transactionを開始するようにサーバへメッセージを送る。このメッセージは、デリートされるワーク、当該トランザクションのためのデリート権のバージョンを示す。

20 ・リポジトリは、共通のオープニングトランザクションステップを実行する。

・サーバはファイルをデリートし、それをファイルシステムから消去する。

・リポジトリは共通のクロージング・トランザクションステップを実行する。

【0182】Directory transaction（ディレクトリ・トランザクション）は、フォルダ、デジタルワーク、及びそれらの部分についての情報に対する要求である。これは、フルパワーの使用権言語のアクセス仕様へ汎用化されることを除いては、TENEXのような従来のファイルシステムにおける保護コードとほぼ同じ考え方である。

【0183】Directory transactionは、デジタルワークに対応している権利及び料金の記述を渡す重要な役割を有している。ユーザが権利を行使したいと思う時、ユーザのリポジトリのユーザインタフェースは、使用可能な権利のバージョンを決定するためにディレクトリ要求を暗黙に作成する。一般に、例えば、権利の行使に対する異なる請求の選択候補を有して、これらはユーザに提供される。このように、多くのDirectory transactionは、ユーザの眼に見えないし、全ての権利を行使することの通常のプロセスの一部として行使される。

・リクエストは、Directory transactionを開始するようにサーバへメッセージを送る。このメッセージはディレクトリ要求の根源であるファイル又はフォルダ、及び当該トランザクションに使用されたディレクトリ権のバージョンを示す。

50 ・サーバはリクエストへの情報がアクセス可能であるこ

とを検証する。特に、これらのディレクトリ仕様においてHIDE-NAME（名前の隠蔽）状態を有するあらゆるファイルのネームをリターンしないし、これらの仕様においてHIDE-PARTS（部分の隠蔽）を有するあらゆるフォルダ又はファイルの部分のリターンしない。情報がアクセス可能でない場合、サーバはエラーでこのトランザクションを終了する。

・リポジトリは、共通のオープニングトランザクションステップを実行する。

・サーバは、転送プロトコルに従って、要求されたデータをリクエストへ送る。

・リクエストはデータを記憶する。

・リポジトリは共通のクロージングトランザクションステップを実行する。

【0184】Folder transaction（フォルダトランザクション）は、フォルダを作成及びリネーム（名前の変更）し、又はワークをフォルダ同士の間で移動するための要求である。Directory rights（ディレクトリ権）と共に、Folder rights（フォルダ権）は、リポジトリの構成が他のリポジトリからアクセス又は修正されることが出来る度合いを制御する。

・リクエストは、サーバへ、Folder transactionを開始するようにメッセージを送る。このメッセージは、フォルダ要求の根源であるフォルダ、及び当該トランザクションのためのフォルダ権のバージョン、動作、及びデータを示す。動作は、ファイルの作成、リネーム、及び移動の内の一つであってもよい。このデータは、フォルダ又はデジタルワーク及びネームの仕様のような動作に対して要求される仕様である。

・リポジトリは、共通のオープニングトランザクションステップを実行する。

・サーバは、要求された動作を実行する。例えば、フォルダ作成、フォルダ・リネーム、又はフォルダ間のワークの移動、を実行する。

・リポジトリは、共通のクロージングトランザクションステップを実行する。

【0185】Extract transaction（抽出トランザクション）は、デジタルワークの一部をコピーし、それを含む新たなワークを作成するための要求である。Extract transaction は、デジタルワークの一部を、追加の制限又は料金が配置されたd-ブロック又はシェルから分離させるために使用され得るという点でコピーとは異なる。抽出動作は、それが、ワークのd-ブロック内への埋め込みだけで、ワークの内容を変更しないという点で編集動作とも異なる。抽出は新しいデジタルワークを作成する。

・リクエストは、サーバへ、Extract transactionを開始するようにメッセージを送る。このメッセージは、抽出されるワークの部分、当該トランザクションにおいて使用される抽出権のバージョン、その部分を新しいワー

クとして配置するための宛て先アドレス情報、ワークのためのファイルデータ、及び含まれるコピー数を示す。

・リポジトリは共通のオープニングトランザクションステップを実行する。

・サーバは、転送プロトコルに従って、要求された内容及びデータをリクエストへ転送する。Next-Set-Of-Rightsが提供された場合、これらの権利が新しいワークに対する権利として転送される。そうでない場合、オリジナルの権利が転送される。この権利に対するCopy-Countフィールドは要求されるコピー数にセットされる。

・リクエストは、内容、データ、及び使用権を記録し、このワークを記憶する。リクエストは、ワークの属性において新しいワークが作成された日時を記録する。

・リポジトリは共通のクロージング・トランザクション・ステップを実行する。

【0186】Embed transaction（埋め込みトランザクション）は、デジタルワークを他のデジタルワークの一部とするか、又はワークのディストリビュータによって料金の追加を可能とさせるためにシェル又はd-ブロックを追加するための要求である。

・リクエストは、Embed transactionを開始するようにサーバへメッセージを送る。このメッセージは、組み込まれるワーク、当該トランザクションに使用される埋め込み権のバージョン、部分をワークとして配置するための宛て先アドレス情報、ワーク用のファイルデータ、及び含まれるコピー数を示す。

・サーバは、部分及び宛て先における全ての権利に対する制御仕様をチェックする。互換性がない場合、サーバはエラーでトランザクションを終了する。

30 ・リポジトリは共通のオープニング・トランザクション・ステップを実行する。

・サーバは、転送プロトコルに従って、要求された内容及びデータをリクエストへ転送する。Next-Set-Of-Rightsが提供された場合、これらの権利がワークに対する権利として転送される。そうでない場合、オリジナルの権利が転送される。この権利に対するCopy-Countフィールドが要求されるコピー数へセットされる。

・リクエストは内容、データ、及び使用権を記録し、宛て先ファイルにワークを組み込む。

40 ・リポジトリは共通のクロージング・トランザクション・ステップを実行する。

【0187】Edit transaction（編集トランザクション）は、現存のデジタルワークの部分のコピー、選択、及び変更することによって新しいデジタルワークを作成するための要求である。この動作は、実際、デジタルワークの内容を変更することができる。許容される変更の種類は、使用されているプロセスに依存する。抽出動作と同様に、編集はデジタルワークの部分上で動作する。抽出動作とは対照的に、編集は、ワークの権利又は位置に影響しない。許容される変更の種類はこれ

らの権利において指定されたプロセッサのタイプ仕様によって決定される。本発明の好ましい実施の形態において、Edit transactionは、ワーク自体を変更し、新しいワークを作成しない。しかしながら、ワークの新しいコピーを作成させるには合理的な変化である。

・リクエストは、サーバへ、Edit transactionを開始するようにメッセージを送る。このメッセージは、編集されるワーク、当該トランザクションに使用される編集権のバージョン、ワーク用のファイルデータ（サイズを含む）、当該プロセスのためのプロセスID、及び含まれるコピー数を示す。

・サーバは、リクエストによって使用されるプロセスIDの権利におけるあらゆるプロセスID仕様に対して互換性をチェックする。互換性がない場合、このトランザクションをエラーで終了する。

・リポジトリは、共通のオープニングトランザクションステップを実行する。

・リクエストは、デジタルワークの内容を所望されるように変更するために当該プロセスを使用する。（例えば、リクエストは、デジタルワークの一部を選択し、複写し、それを他の情報と連結し、又は情報に基づいて関数を演算することができる。これは、結局、テキスト、音楽、又は画像（絵）を編集すること、又はデリバティブ（系列）ワークを作成するのに有用な他のステップを任意に使うことに相当する。）

・リポジトリは、共通のクローニング・トランザクション・ステップを実行する。

【0188】Edit transaction（編集トランザクション）は、広範囲の種類ワークを網羅するために使用される。デジタルワークの任意の部分を入力として取り、次に、何らかの方法でこの入力を変更するプロセスが該トランザクションのカテゴリである。例えば、テキストに関しては、このテキストを編集するためのプロセスは、編集権を要求する。テキスト中のワードを「要約」し、又はカウントするプロセスも編集と見なされる。音楽ファイルに関しては、プロセッシングは、ピッチやテンポの変更、若しくは反響（エコー）の追加、又はあらゆる他のオーディオ効果を含むことができる。デジタルビデオに関しては、画像を変更するものは編集権を必要とする。例としては、着色、スケーリング（拡大縮小）、スチール写真の抽出、フレームを選択し結合してストーリーボードを作成すること、信号プロセッシングによる画像の鮮明化、その他である。

【0189】何人かのクリエイターは、（ワーク上で）実行されるプロセスの種類を制限することによって彼らのワーク（作品）の認証を保護したがるかもしれない。編集権がない場合、プロセッシングは全く許容されない。どの種類のプロセスが許容されるかを指定するためにプロセッサ識別子が含まれることができる。プロセッサ識別子が指定されなかった場合、任意のプロセッサが使用

され得る。特定プロセスの一例に関しては、写真家は彼の写真の使用をオーソライズしようとするかもしれないが、着色されるのをいやがるかもしれない。音楽家は彼の作品の部分の抽出をオーソライズしようとするかもしれないが、階調特性を変更するのはいやがるかもしれない。

【0190】オーソライゼーショントランザクションが定義されるには多数の方法がある。以下において、一つの好ましい方法は、我々がリポジトリのために既に必要とする他のトランザクションに関してそれらを簡単に定義することである。このように、「オーソライゼーショントランザクション」を述べるのはしばしば簡単ではあるが、これらのトランザクションは実際にはリポジトリが既に有している他のトランザクションから構成されている。

【0191】使用権は、オーソライゼーションオブジェクト（標準フォーマットファイルにおけるデジタルワーク）を識別するオーソライゼーションIDを指定することができ、このオーソライゼーションオブジェクトは、リポジトリが所有していなければならないと共に当該リポジトリが処理しなければならない。オーソライゼーションは、当該オーソライゼーションの解釈を開始するリポジトリのジェネリックオーソライゼーション（又はチケット）サーバへ付与される。

【0192】前に述べたように、オーソライゼーションはサーバ識別子を含み、該サーバ識別子はジェネリックオーソライゼーションサーバ又は他のサーバであってもよい。遠隔オーソライゼーションサーバが要求される時、サーバはデジタルアドレスを含むことができる。当該サーバはさらにデジタル証明書を含むことができる。

【0193】遠隔オーソライゼーションサーバが要求される場合、オーソライゼーションプロセスは最初に以下のステップを実行する。

・ジェネリックオーソライゼーションサーバは、通信チャネルをセットアップ（設定）しようとする。（チャネルがセットアップされない場合、オーソライゼーションはエラーで失敗する。）

・チャネルがセットアップされた場合、遠隔リポジトリによって登録プロセスを実行する。（登録に失敗した場合、オーソライゼーションはエラーで失敗する。）

・登録が終了すると、ジェネリックオーソライゼーションサーバは遠隔リポジトリによって「再生」トランザクションを呼び出し、再生されるデジタルワークとしてのオーソライゼーションドキュメントと、「プレーヤー」としての遠隔オーソライゼーションサーバ（プログラム）を供給する。（プレーヤーが見つからないか又はプレーヤーがいくつかの他のエラーを有していた場合、オーソライゼーションはエラーで終了する。）

・オーソライゼーションサーバは、次に、オーソライゼ

ーションを「再生」する。これは、証明書を発行したマスターポジトリのパブリックキー又はそれを転送したりポジトリからのセッションキーのいずれかを使ってそれを復号することを含む。オーソライゼーションサーバは、種々のテストを実行する。これらのテストはオーソライゼーションサーバによって変化する。これらは、オーソライゼーションの発行及び有効期間をチェックし、知られている無効オーソライゼーションのホットリストをチェックするようなステップを含む。オーソライゼーションサーバは、リポジトリ上においても、ディレクトリをチェックし、パスワードを送ってくれる誰かを見つけ、又は幾つかの他のデジタルワークの再生のようなあらゆる他のトランザクションを実行することを要求することができる。オーソライゼーションサーバは、さらに、位置又は最近の事象に関する情報をチェックするためにいくつかの特別なプロセスを呼び出してもよい。このようなステップに関する「スクリプト」はオーソライゼーションサーバ内に含まれている。

・要求されるステップの全てが申し分なく終了した場合、オーソライゼーションサーバはトランザクションを正常に終了し、このオーソライゼーションが許諾されたことを合図する。

【0194】 Install transaction (インストール・トランザクション) はリポジトリ上で実行可能なソフトウェアとしてデジタルワークをインストールするための要求である。一般的なケースにおいて、リクエスト・リポジトリはレンダリングリポジトリであり、このソフトウェアは新しい種類又は新しいバージョンのプレーヤーである。さらに一般的なケースにおいて、ソフトウェアは、リクエストリポジトリがインストールされる前に、当該リクエストリポジトリのファイルシステムにコピーされる。

・リクエストはサーバへ Install メッセージを送る。このメッセージは、インストールされるワーク、呼び出されるインストール権のバージョン、及びそのワークのためのファイルデータ (そのサイズを含む) を指示する。

・リポジトリは、共通のオープニングトランザクションステップを実行する。

・リクエストは、ソフトウェアのためのデジタル証明書のコピーを抽出する。この証明書が見つからず、又はこの証明書に対するマスターポジトリがリクエストに知らされていない場合、トランザクションはエラーで終了する。

・リクエストは、マスターポジトリのパブリックキーを用いてデジタルワーク証明書を復号すると共に、供給者とクリエータの識別、ソフトウェアを復号するためのキー、互換性情報、及び偽造チェックコードを記録する。(このステップはソフトウェアを認証する。)

・リクエストは、証明書からのキーを用いてソフトウェアを復号し、単方向ハッシュ関数を用いてチェックコー

ドをリクエスト上で演算する。チェックコードが証明書からの偽造チェックコードと符号しない場合、インストール・トランザクションはエラーで終了する。

(このステップは、種々のスクリプトを含むソフトウェアの内容が偽造されていないことを確認とする。)

・リクエストは互換性チェックスクリプトにおいて命令を検索し、それらをフォローする。ソフトウェアがリポジトリと互換性がない場合、インストール・トランザクションはエラーで終了する。(このステップはプラットフォームの互換性をチェックする。)

・リクエストはインストールスクリプトにおいて命令を検索し、それらをフォローする。このプロセスにおいてエラーがあった場合 (例えば、不十分なりソース)、トランザクションはエラーで終了する。インストールプロセスは、他のトランザクションを実行する時のリポジトリ動作の一部としてソフトウェアを実行すること以外の任意の使用権を行使するためのワークとしてもはやアクセスできないリポジトリに、実行可能なソフトウェアを配置することに注目されたい。

20 ・リポジトリは共通のクロージングトランザクションステップを実行する。

【0195】 Uninstall transaction (アンインストール・トランザクション) は、リポジトリからソフトウェアを除外するための要求である。リポジトリからのソフトウェアの制御されない又は不正確な除外がその行動上の完全性を損傷し得るので、このステップは制御される。

・リクエストは、Uninstall メッセージをサーバへ送る。このメッセージは、インストールされないワーク、

30 呼び出される Uninstall 権のバージョン、及びこのワークのためのファイルデータ (そのサイズを含む) を示す。

・リポジトリは、共通のオープニングトランザクションステップを実行する。

・ソフトウェアのためのデジタル証明書のコピーを抽出する。この証明書が見つからず、又はこの証明書のためのマスターポジトリがリクエストに知らされていない場合、トランザクションはエラーで終了する。

・リクエストは、ソフトウェアがインストールされたか否かをチェックする。ソフトウェアがインストールされない場合、トランザクションはエラーで終了する。

・リクエストは、マスターリクエストのパブリックキーを用いてデジタル証明書を復号すると共に、供給者とクリエータの識別、ソフトウェアを復号するためのキー、互換性情報、及び偽造チェックコードを記録する。(このステップは、ソフトウェアを除外するためのスクリプトを含むソフトウェアの証明書を認証する。)

・リクエストは、証明書からのキーを用いてソフトウェアを復号し、単方向ハッシュ関数を用いてチェックコードをリクエスト上で演算する。チェックコードが証明書

からの偽造チェックコードと符号しない場合、インストール・トランザクションはエラーで終了する。

(このステップは、種々のスクリプトを含むソフトウェアの内容が偽造されていないことを確認とする。)

・リクエストは、アンインストール（除外設定）スクリプトにおける命令を検索し、これらの命令をフォローする。このプロセスにエラーが発生した場合（例えば、不十分なリソース）、当該トランザクションはエラーで終了する。

・リポジトリは共通のクロー징・トランザクション・ステップを実行する。

【0196】

【発明の効果】 デジタルワークの使用及び配給を制御するシステムを提供する。

【図面の簡単な説明】

【図1】 本発明の好ましい実施の形態の動作の単純なインスタンス生成を示すフローチャートである。

【図2】 本発明の好ましい実施の形態において、種々のリポジトリのタイプを示し、リポジトリ同士間でのリポジトリ・トランザクション・フローを示すブロック図である。

【図3】 本発明の好ましい実施の形態において、クレジットサーバに連結されたリポジトリのブロック図である。

【図4】 図4の（a）は、本発明の実施の形態において使用され得るレンダリングシステムの実施例を示す図である。図4の（b）は、本発明の実施の形態において使用され得るレンダリングシステムの実施例を示す図である。

【図5】 本発明の好ましい実施の形態において使用され得るデジタルワークに対する内容ファイルレイアウトを示す図である。

【図6】 本発明の好ましい実施の形態において使用され得る図5のデジタルワークの個々のデジタルワークに対する内容ファイルレイアウトを示す図である。

【図7】 本発明の好ましい実施の形態の記述ブロックの構成要素を示す図である。

【図8】 図5に示されたデジタルワークの内容ファイルレイアウトに対する記述ツリーを示す図である。

【図9】 図6に示された個々のデジタルワークに対応している記述ツリーの部分を示す図である。

【図10】 本発明の実施の形態において使用される記述

ブロックの権利部分に対するレイアウトを示す図である。

【図11】 いくつかのd-ブロックがPRINT 使用権を有する記述ツリーであり、当該記述ツリーが使用権の競合を解消するための「厳格」及び「寛大」なルールを示すために使用される図である。

【図12】 本発明の好ましい実施の形態において使用されるリポジトリのハードウェア構成要素のブロック図である。

10 【図13】 本発明の好ましい実施の形態において使用されるリポジトリの機能的（論理的）構成要素のブロック図である。

【図14】 本発明の好ましい実施の形態における使用権の基本構成要素を示すブロック図である。

【図15】 本発明の好ましい実施の形態の使用権文法のプログラム表示例である。

20 【図16】 本発明の好ましい実施の形態で実行されるように、登録トランザクションにおいて実行される証明書デリバリ、ホトリストチェック及びパフォーマンステストのステップを示すフローチャートである。

【図17】 登録トランザクションにおける各リポジトリが図16に示したステップを無事完了した後で、本発明の好ましい実施の形態において実行されるセッション情報交換とクロック同期のステップを示すフローチャートである。

【図18】 本発明の好ましい実施の形態で実行されるように、共通のオープニング及びクロージングステップを含む使用トランザクションに対する基本フローを示すフローチャートである。

30 【図19】 本発明の好ましい実施の形態で実行されるように、デジタルワークをサーバからクライアントリポジトリへ移動する時にフォローされる伝送プロトコルに従ったサーバ及びクライアントリポジトリの状態のブロック図である。

【符号の説明】

201 リポジトリ

301 クレジットサーバ

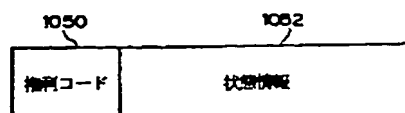
401 プリントシステム

402 プリンタリポジトリ

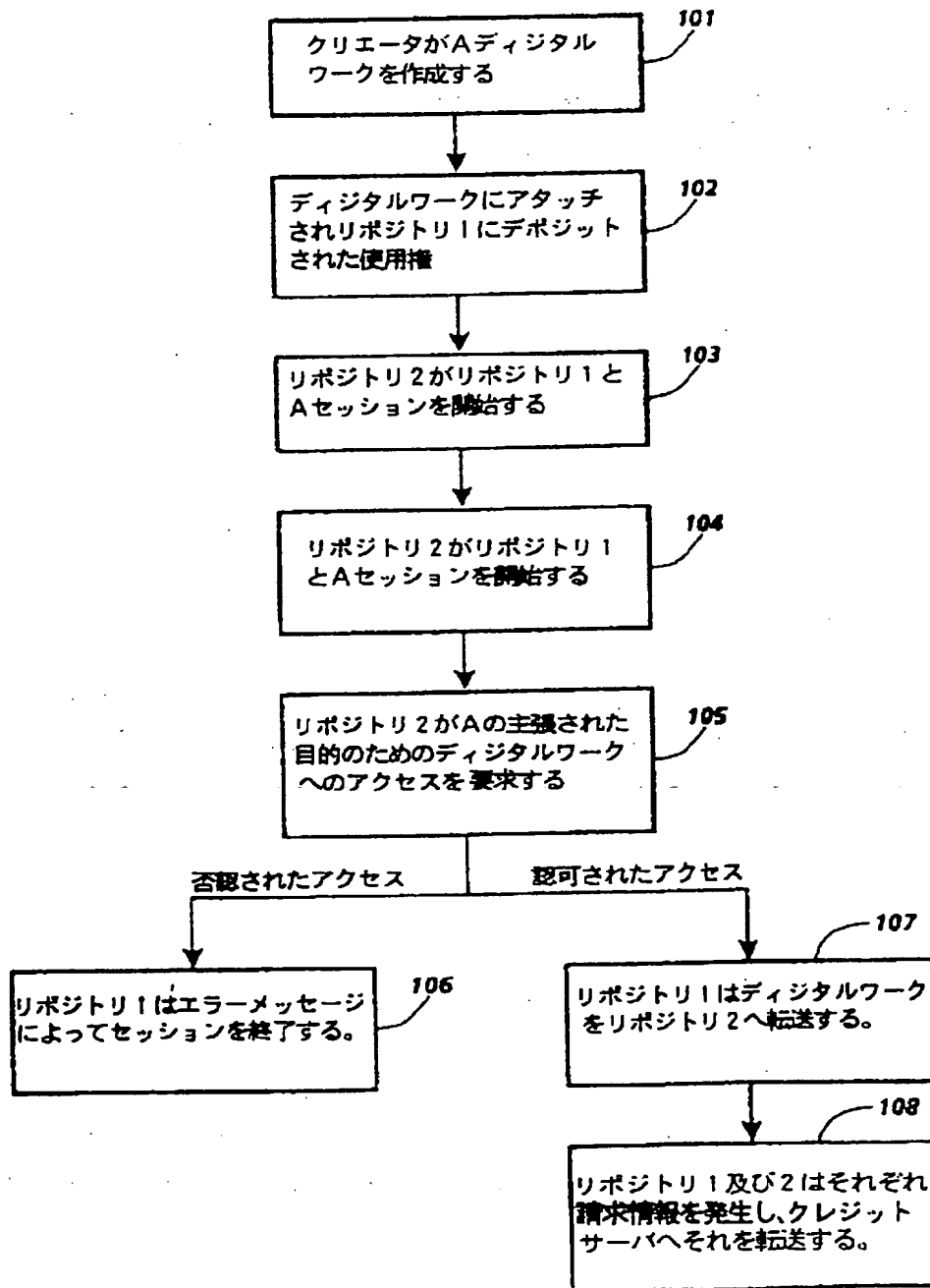
40 403 プリントデバイス

410 マルチ機能システム

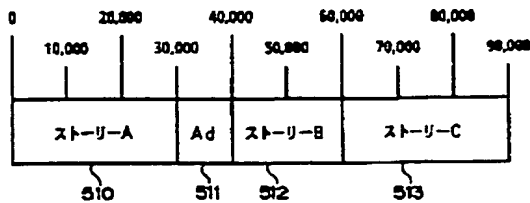
【図10】



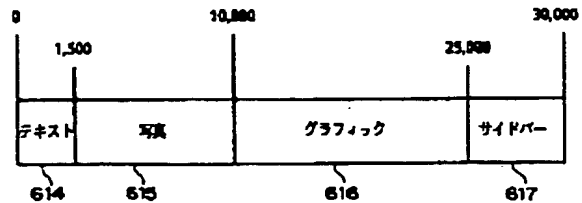
【図1】



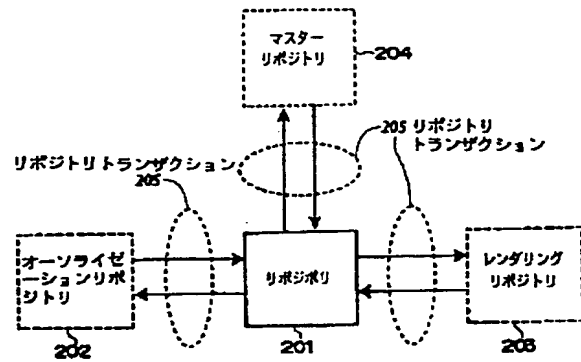
【図5】



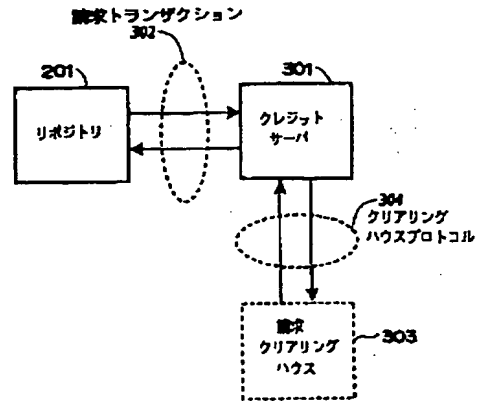
【図6】



【図 2】

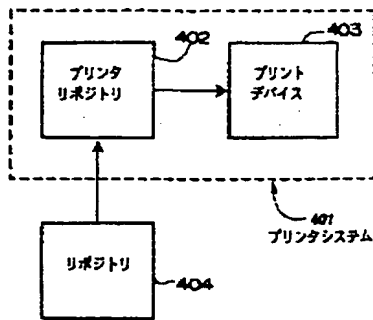


【図 3】

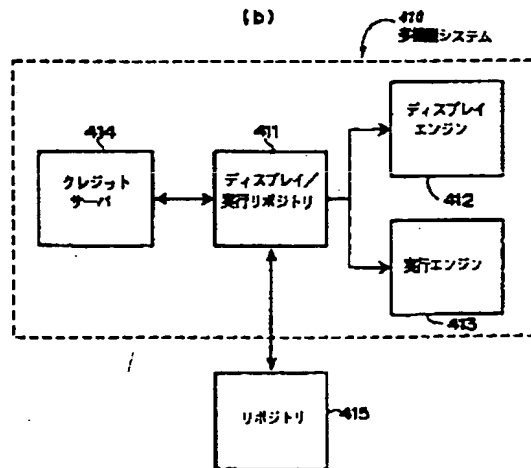


【図 4】

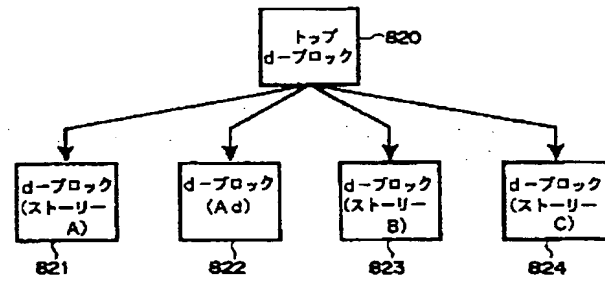
(a)



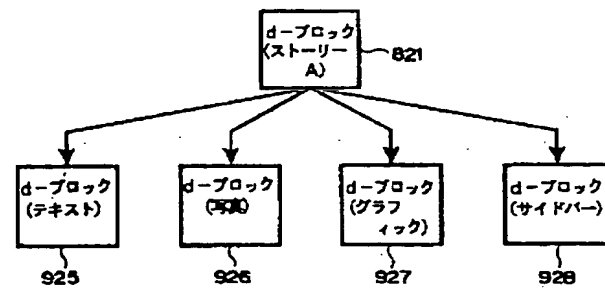
(b)



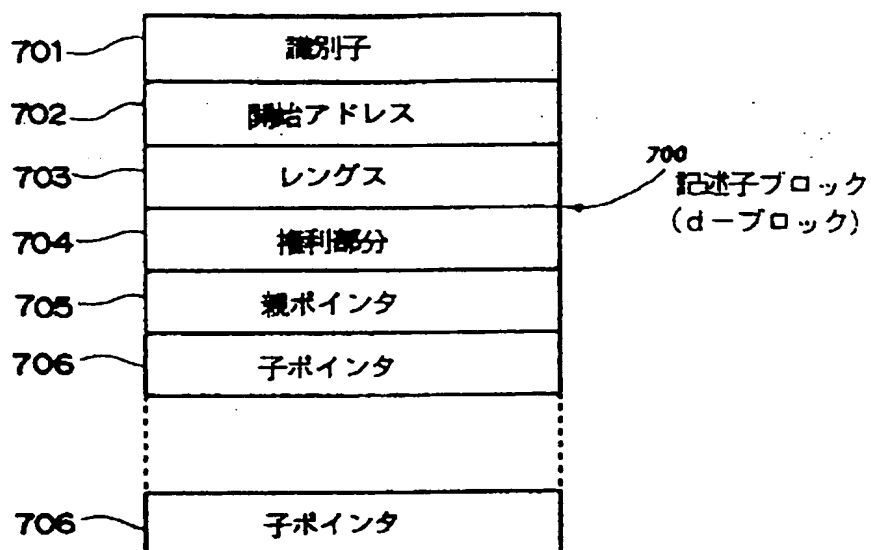
【図 8】



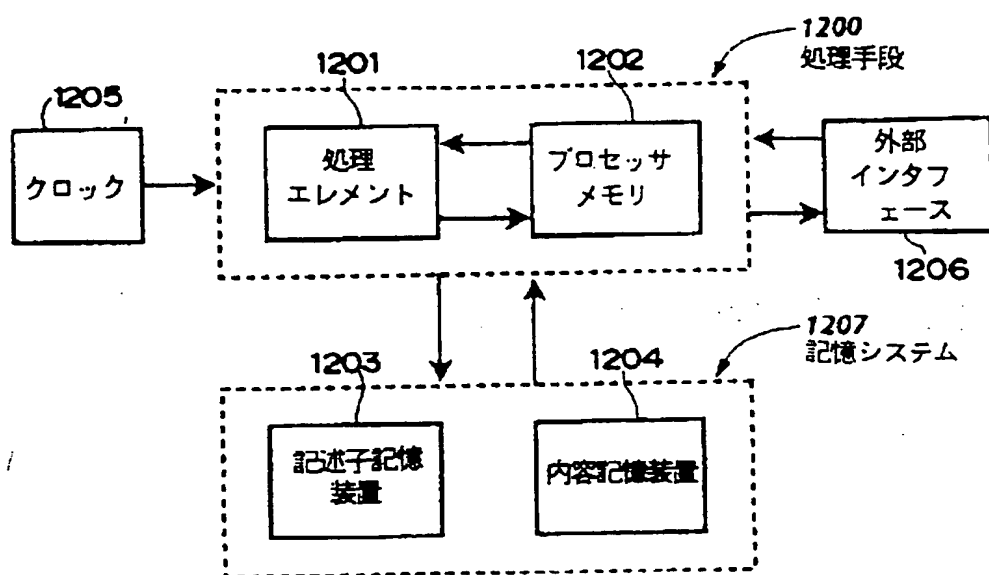
【図 9】



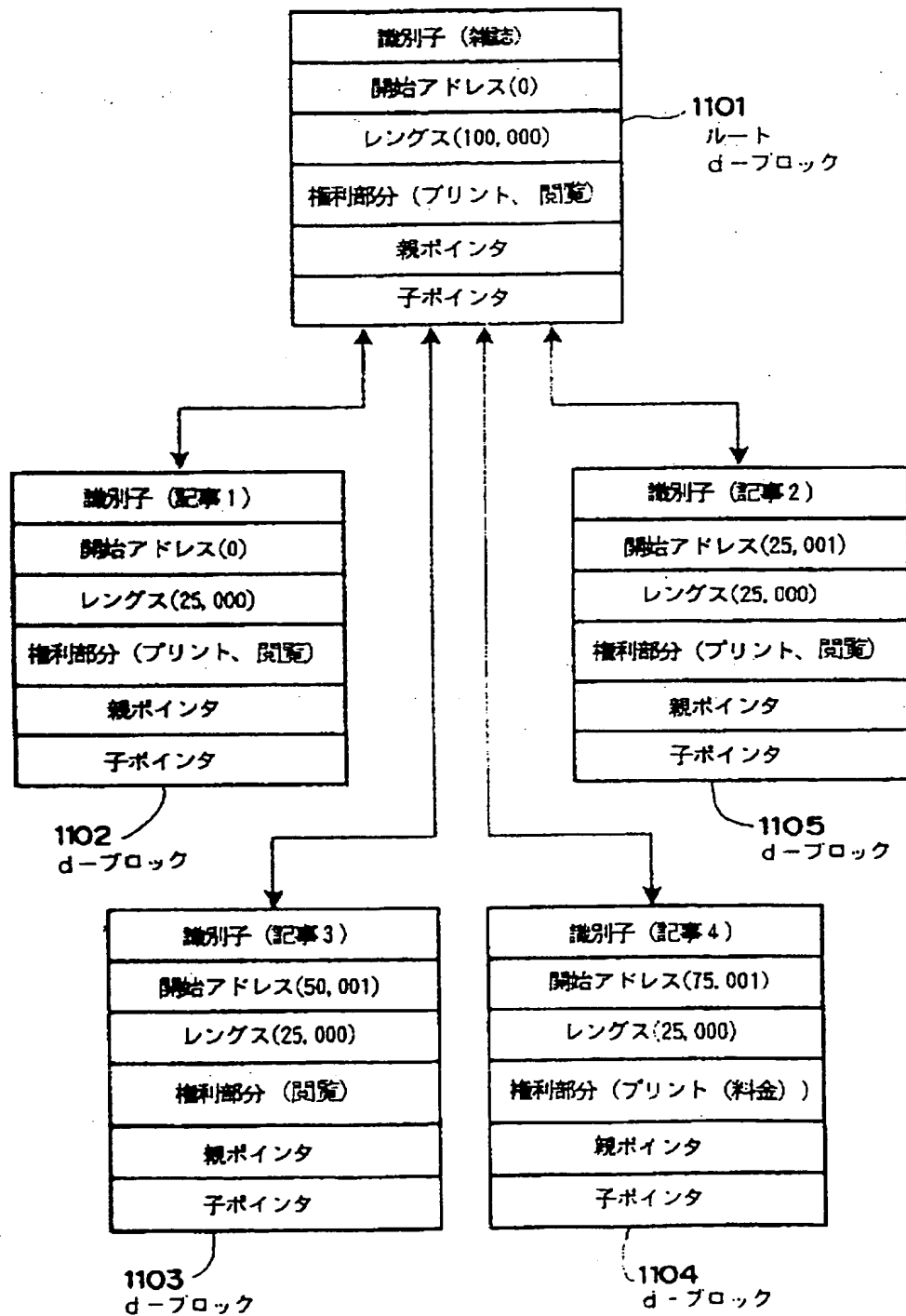
【図7】



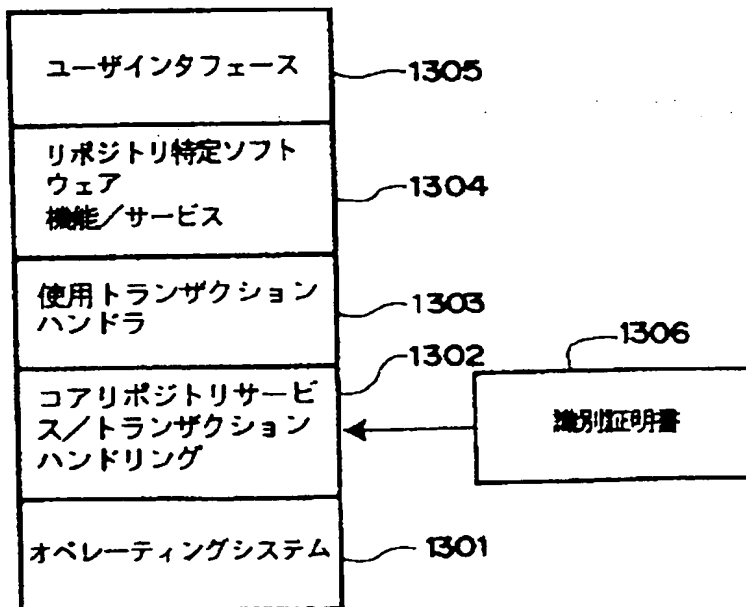
【図12】



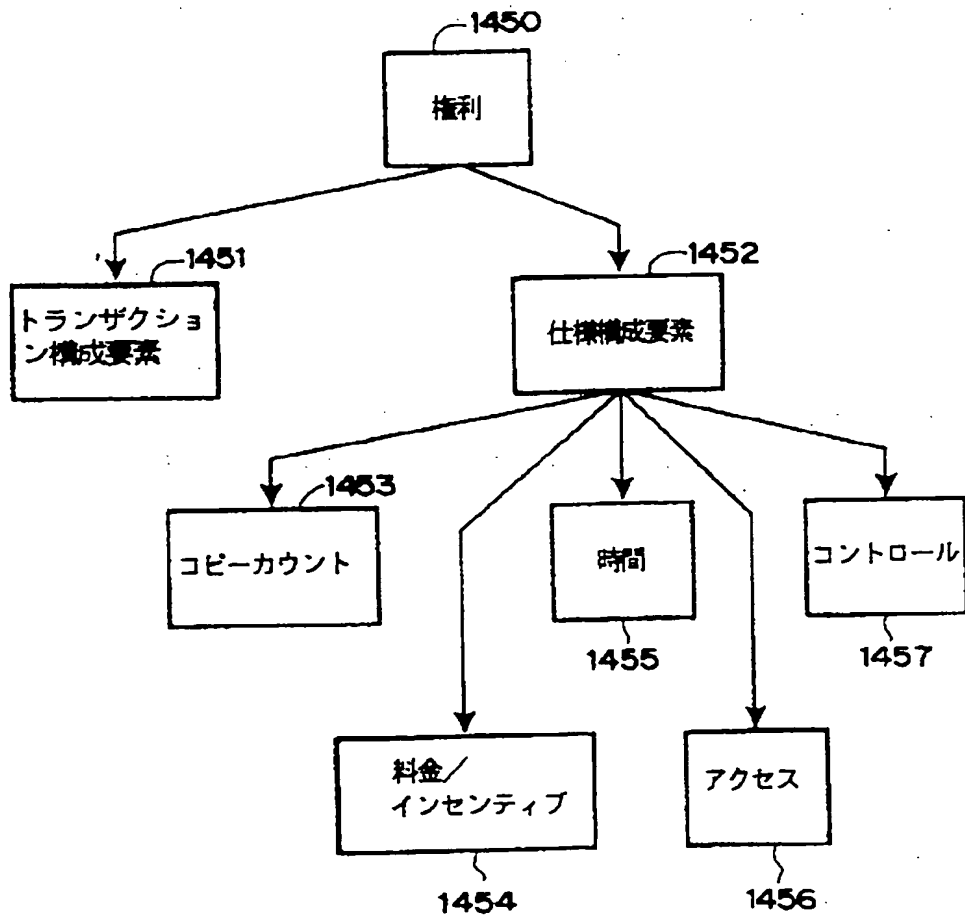
【図11】



【図13】



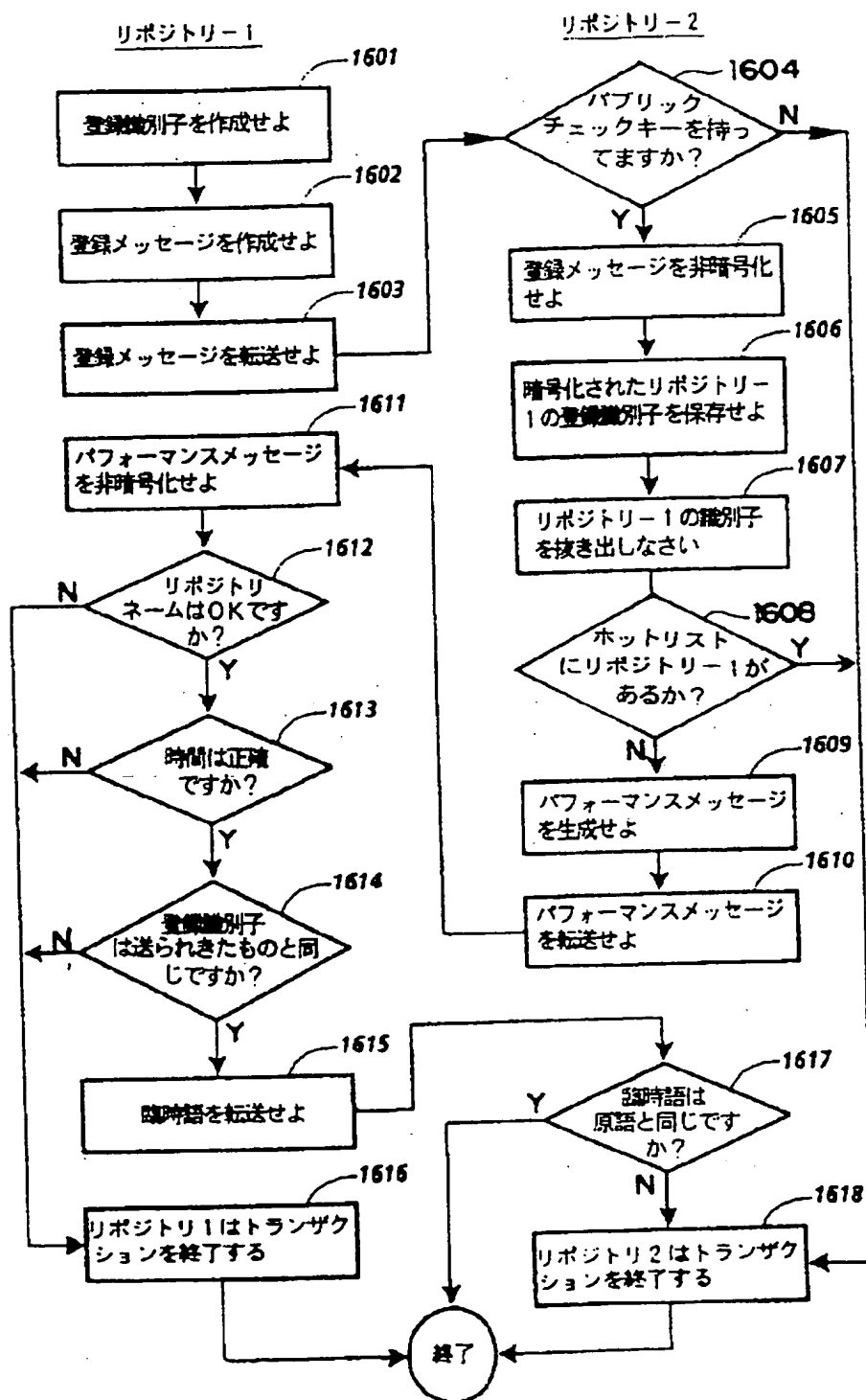
【図14】



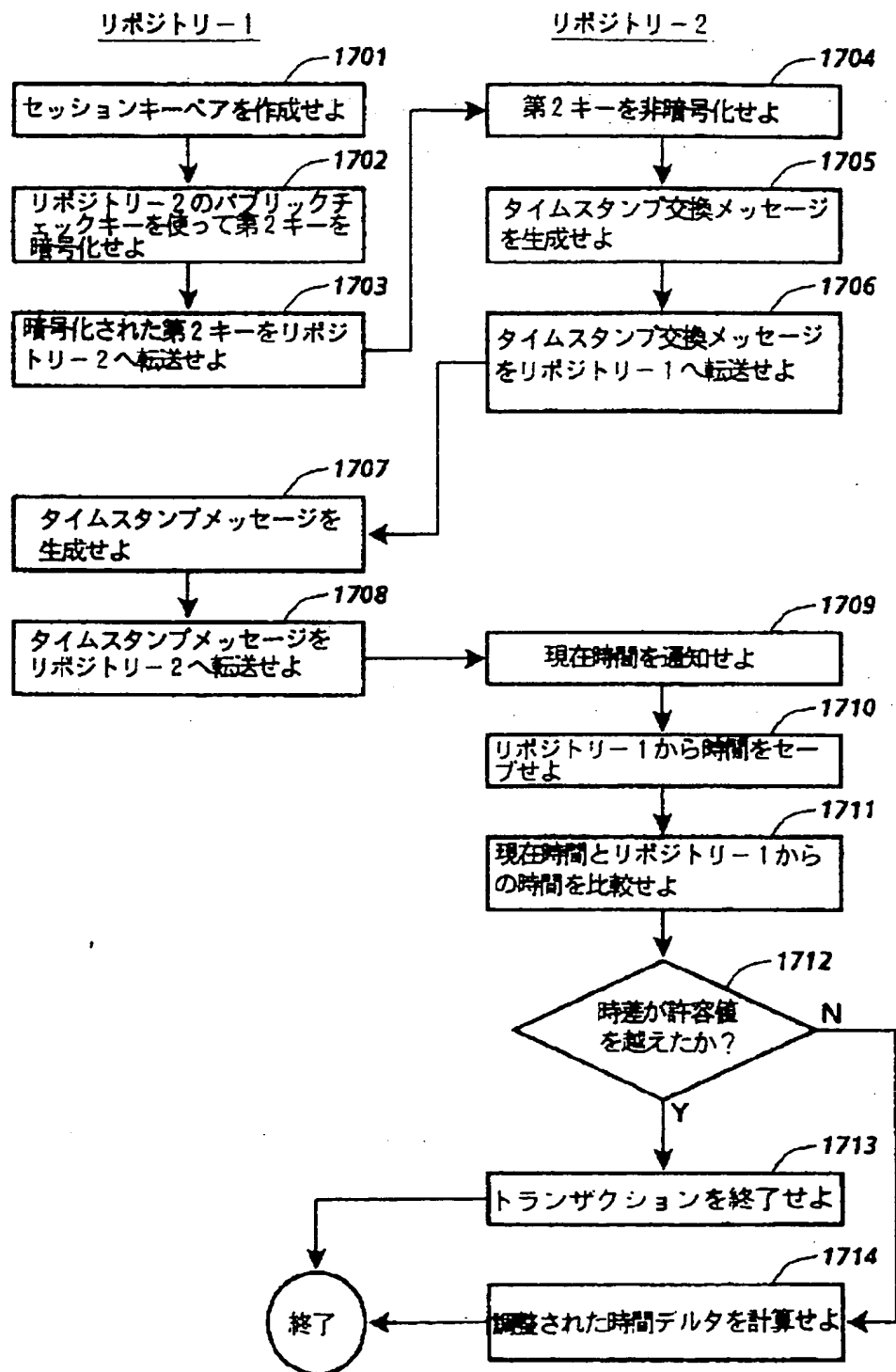
【図15】

1501 ~ Digital Work Rights := "Rights"
 1502 ~ Right := (Right-Code (Copy-Count) (Control-Spec) (Time-Spec)
 (Access-Spec) (Fee-Spec))
 1503 ~ Right-Code := Render-Code | Transport-Code | File-Management-
 Code | Derivative-Works-Code | Configuration-Code
 1504 ~ Render-Code := (Play (Player: Player-ID) | Print (Printer: Printer-ID))
 1505 ~ Transport-Code := (Copy | Transfer | Loan (Remaining-Rights:
 Next-Set-of-Rights)) | (Next-Copy-Rights: Next-Set-of-Rights)
 1506 ~ File-Management-Code := Backup (Back-Up-Copy-Rights:
 Next-Set-of-Rights) | Restore | Delete | Folder
 | Directory (Name: Hide-Local | Hide-Remote)
 | Parts: Hide-Local | Hide-Remote
 1507 ~ Derivative-Works-Code := (Extract | Embed | Edit (Process:
 Process-ID)) | (Next-Copy-Rights:
 Next-Set-of-Rights)
 1508 ~ Configuration-Code := Install | Uninstall
 1509 ~ Next-Set-of-Rights := ((Add: Set-Of-Rights) | (Delete:
 Set-Of-Rights) | (Replace: Set-Of-Rights) | (Keep: Set-Of-Rights))
 1510 ~ Copy-Count := (Copies: positive integer | 0 | Unlimited)
 1511 ~ Control-Spec := (Control: (Restrictable | Unrestrictable)
 (Unchargeable | Chargeable))
 1512 ~ Time-Spec := ((Fixed-Interval | Sliding-Interval | Meter-Time)
 Until: Expiration-Date)
 1513 ~ Fixed-Interval := From: Start-Time
 1514 ~ Sliding-Interval := Interval: Use-Duration
 1515 ~ Meter-Time := Time-Remaining: Remaining-Use
 1516 ~ Access-Spec := ((SC: Security-Class) | (Authorization: Authorization-ID*)
 (Other-Authorization: Authorization-ID*) | (Ticket: Ticket-ID))
 1517 ~ Fee-Spec := (Scheduled-Discount) | Regular-Fee-Spec | Scheduled-Fee-Spec |
 Markup-Spec
 1518 ~ Scheduled-Discount := Scheduled-Discount: (Scheduled-Discount:
 (Time-Spec Percentage)*)
 1519 ~ Regular-Fee-Spec := ((Fee: (Incentive:) | (Per-Use-Spec | Metered-Rate-
 Spec) | Best-Price-Spec | Call-For-Price-Spec)
 (Min: Money-Unit Per: Time-Spec) | (Max:
 Money-Unit Per: Time-Spec) | To: Account-ID)
 1520 ~ Per-Use-Spec := Per-Use: Money-unit
 1521 ~ Metered-Rate-Spec := Metered: Money-Unit Per: Time-Spec
 1522 ~ Best-Price-Spec := Best-Price: Money-unit Max: Money-unit
 1523 ~ Call-For-Price-Spec := Call-For: Price
 1524 ~ Scheduled-Fee-Spec := (Schedule: (Time-Spec Regular-Fee-Spec)*)
 1525 ~ Markup-Spec := Markup: percentage To: Account-ID

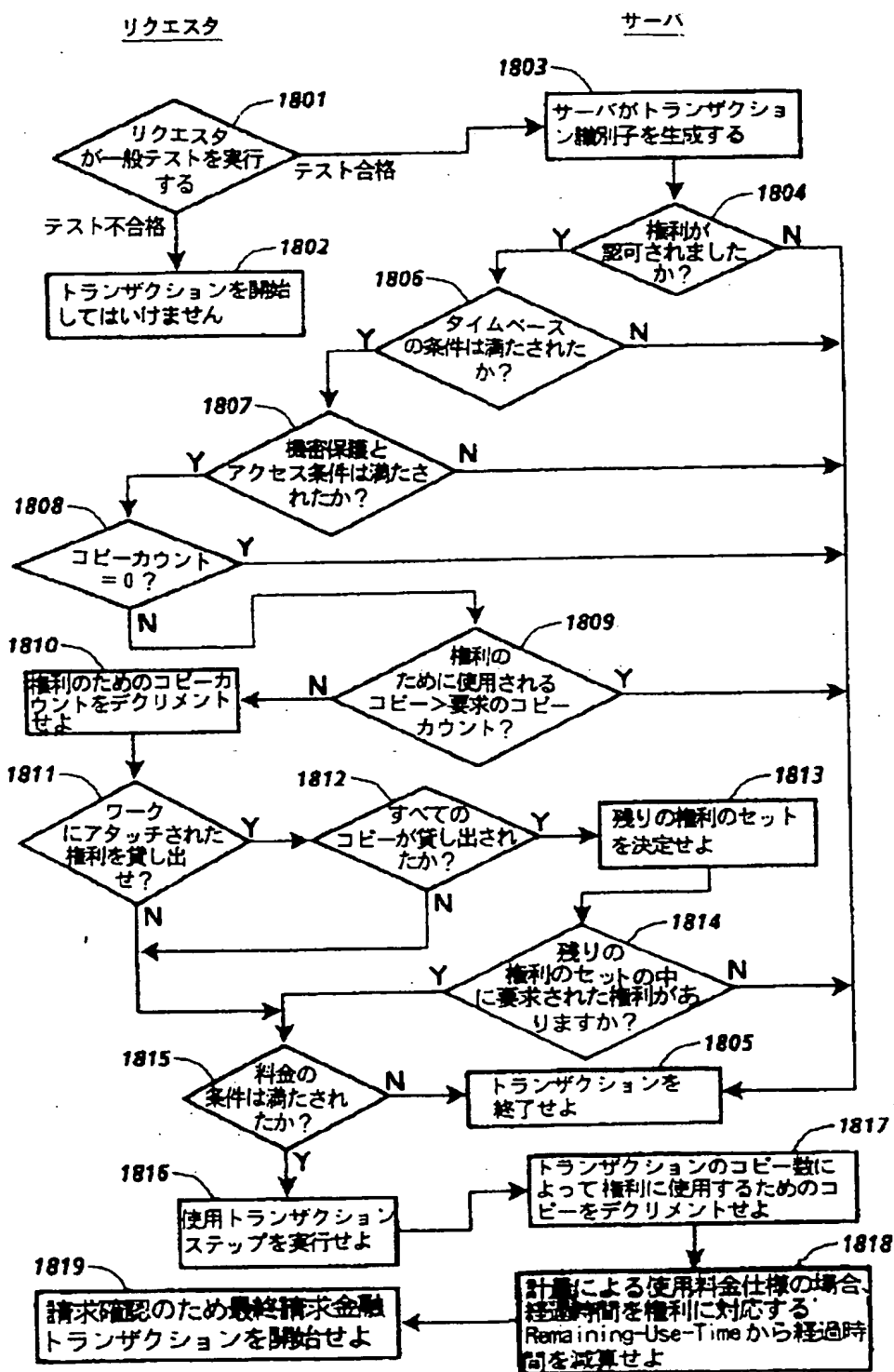
【図16】



【図17】



【図18】



```

sequenceDiagram
    participant Server as サーバ
    participant Client as クライアント
    Note over Server: 1901 ライン
    Server->>Server: 1902 新しいトランザクション
    Server->>Server: 1903 スタート
    Server->>Server: 1906 次のデータを送りなさい
    Server->>Server: 1907 データ
    Server->>Server: 1908 Ackを待機せよ
    Server->>Server: 1909 データはもういない
    Server->>Server: 1910 アクノレッジ
    Server->>Server: 1912 (キャンセル) 失敗
    Server->>Server: 1914 クレジットサーバへ
    Server->>Server: 1915 クレジットサーバへエラー報告せよ
    Server->>Server: 1916 実行されました

    Note over Client: 1904 トランザクションを待機せよ
    Client->>Client: 1905 データを待機せよ
    Client->>Client: 1909 受け取ったデータ
    Client->>Client: 1910 アクノレッジ
    Client->>Client: 1913 (キャンセル) 失敗
    Client->>Client: 1916 クレジットサーバへ
    Client->>Client: 1918 クレジットサーバへエラー報告せよ
    Client->>Client: 1919 実行されました

    Server-->>Client: 1903
    Client-->>Server: 1907
    Server-->>Client: 1909
    Client-->>Server: 1910
    Server-->>Client: 1912
    Client-->>Server: 1913
    Server-->>Client: 1914
    Client-->>Server: 1915
    Server-->>Client: 1916
    Client-->>Server: 1918
    Server-->>Client: 1919
  
```

6 7 3 Z